INTRUSION DETECTION IN MOBILE AD-HOC NETWORKS IMPLEMENTATION AND PERFORMANCE EVALUATION OF ADAPTIVE ACKNOWLEDGMENT APPROACH

BY

ANAS ABDULWAHED HASAN AL-ROUBAIEY

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

JUNE 2009

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ANAS ABDULWAHED HASAN AL-ROUBAIEY** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** in **COMPUTER NETWORKS**

Thesis Committee

Dr. Tarek Sheltami (Advisor)

Dr. Adnan Gutub (Member)

Dr. Ashraf Mahmoud (Member)

Dr. Adnan A. Gutub
(Department Chairman)

Dr. Salam A. Zummo
(Dean of Graduate Studies)

24/6/09
Date

إهـــــــــــداء

أهدي هذا العمل المتواضع إلى والدي حفظهم الله ورعاهم لصبرهم و دعائهم و تشجيعهم ولما قدماه لي من جهد و دعم مادي ومعنوي متواصل خلال مسيرتي العلمية , و مهما عملت فلن استطيع أن أفي ولو بالجزء اليسير مما قدماه , اسأل الله العظيم رب العرش العظيم أن يلبسهم تاج الصحة والعافية و أن يوفقني إلى طاعتهم و برهم. كما أهدية إلى ينبوع الصبر والأمل إلى زوجتي أم عبد الجليل و إلى أخواتي و إخواني لدعمهم لي بالتشجيع والدعاء.

ACKNOWLEDGMENT

All the praises and thanks be to ALLAH, the Almighty, whose blessing and help are all the time with me, and He is the only one Who helps.

I deeply thank my advisor, Dr. Tarek Sheltami, for his unlimited support, help, advise and encouragement. Thanks are extended to my committee members Dr. Adnan Gutub and Dr. Ashraf Mahmoud for their valuable comments, support and guidance throughout this research

Great thanks due to King Fahd University of Petroleum and Minerals represented by Dr. Khaled Al-Sultan for supporting me throughout this research and during my MS study.

ABSTRACT

Anas A. Al-Roubaiey

King Fahd University of Petroleum and Minerals, 2009

A mobile ad-hoc network (MANET) is an infrastructureless network consisting of self-configuring mobile nodes connected by wireless links. Because of its decentralized property, these nodes relay on each other to store and forward packets. Most of the proposed MANET protocols assume cooperative and friendly network context, and do not address security issues. Furthermore, MANETs are highly vulnerable for passive and active attacks because of their open medium, rapidly changing topology, lack of centralized monitoring. The encryption and authentication solutions, which are considered as the first line of defense, are not sufficient to protect MANETs. Therefore, Intrusion Detection Systems (IDSs) are needed to be the second line of defense to protect the network from such security problems. Many of the current IDSs for MANETS based on the Watchdog technique. In this research we study the behavior of this technique and propose a novel mechanism, which named as Adaptive ACKnowledgment (AACK), for solving two significant problems of this technique, which are the limited transmission power and receiver collision. We use NS2 to simulate our scheme and compare it with the existing schemes such as TWOACK. Video traffic is used as a real application to evaluate our scheme. Our results show that AACK outperforms watchdog and TWOACK in term of network packet delivery ratio.

ملخص الرسالة

أكتشاف التطفل في شبكات الأدهوك المتحركة ـ تنفيذ وتحليل ادآء طريقة الآدابتف اكنولوجمينت

الاسم: أنس عبدالواحد الربيعي

القسم: هندسة الحاسب الالي

التخصص: شبكات الحاسب الآلي

تاريخ المناقشة: 6\6\2009

شبكة الأدهوك المتحركة هي شبكة لاسلكية بدون بنية تحتية تتكون من أجهزة (طرفيات) لاسلكية متحركة تعتمد ذاتيا على ملاءمة اعداداتها للتواصل فيما بينها و تكوين شبكة الأدهوك . و نتيجة لعدم وجود ادارة مركزية فأن هذه الأجهزة تتكل على بعضها البعض في نقل بيانات الشبكة . معظم البروتوكولات المقدمة لنقل البيانات في شبكة الأدهوك تفترض وجود بيئة صديقة و متعاونة ,و تفتقر لأي احتياطات امنية لتأمين نقل البيانات . علاوة على ذلك فإن هذا النوع من الشبكات لديه ثغرات عديدة تجعلها عرضة لأنواع متعددة من الهجمات العدوانية , و ذلك بسبب عملها في بيئة مفتوحة , و سرعة تغيير مواقع أجهزتها , و افتقارها للتحكم المركزي . تقنيات التشفير و تحديد الهوية , التي تعتبر خط الدفاع الأول , ليست كافية لحماية الشبكة من المتطفلين . لذلك تحتاج الشبكة لخط دفاع ثاني وهو انظمة اكتشاف التطفل التي تعتبر مكملة لأنظمة التشفير و تحديد الهوية لحماية الشبكة من هجمات المتطفل ين. هذا البحث يركز على الهجمات التي تتعمد اسقاط البيانات بدلا من توصيلها. كثير من الحلول المقدمة ضد هذا النوع من الهجمات تعتمد على تقنية الووتش دوج (Watchdog). يدرس هذا البحث سلوك الووتش دوج و يقدم تقنية جديدة تزيد من كفاءة الووتش دوج تعرف بإسم ادابتف آك (Adaptive ACK) لتحل مشكلتين رئسيتين لتقنية الووتش دوج . هذه المشاكل هي تصادم البيانات عند المستقبل , و التحكم بطاقة الارسال للتغرير على المرسل . استخدمنا في هذا البحث برنامج المحاكاة الشهير الإن إس تو (NS-2) لمحاكاة حلنا المقدم و قارناه بالحل المودجود التو آك (TWOACK). النتائج دلت على ان الادابتف آك اكثر كفاءة من الووتش دوج و التو آك و كذلك اقل تكلفة من التو آك.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

Before describing the fundamental research topic, a brief introduction into Mobile Ad-hoc NETworks (MANETs) is introduced to make the reader more familiar with the concept of ad-hoc networks.

1.1 Ad-hoc Networks Overview

Ad-hoc is a Latin expression which means "for this purpose". Unlike conventional cellular wireless networks that need an expensive infrastructure to support mobility, MANETs do not need the expensive or wired infrastructure. In some situations, the traditional wireless network that needs a fixed network infrastructure, with base stations or access points, is not applicable or not suitable to be used (e.g., military missions, disaster recovery or temporary networks like conferences). In such situations we need a fast deployment and self-organized network that will be used just for a specific purpose for specific period of time. Ad-hoc networks are decentralized, self-configuring, self-organizing networks and are capable of forming a communication network without relying on any fixed infrastructure. They are composed of nodes that relay on each other to manage and forward their traffic. Therefore, all the nodes are working as a usual node and as a router, where they route the packets of the other nodes. Wireless ad-hoc networks can by classified into two types: single hope and multi-hop networks. When there are no intermediate nodes between the source and the destination we call the network a single hope network. For example, a laptop communicates with other devices like a PDA or a video camera using Bluetooth. On the other hand, if the source relay on the other nodes to transmit its packets to the destination the network is called a multi-hop

ad-hoc network which makes the nodes that are out of radio range of each other to be communicating using the help of the other intermediate nodes. A High level description about the ad-hoc networks and their related researches is in [Perkins 2000] [Ilyas 2002] [Hekmat 2006] [Barbeau 2007][Sarkar 2008].

There are many applications for MANETs. Originally, they were developed for military purposes, for example, as nodes are scattered on a battlefield for surveillance mission [Xiao 2006] and connectivity beyond the line of sight. In recent years, the use of MANETs is rapidly increased and ranging from military to civilian and commercial uses. For example, we can think of a group of people in conference, they use ad-hoc networks to communicate with each other to exchange files and data by using their laptops or PDAs. Another promising example for ad-hoc applications is the ubiquitous computing like the communication between the smart household appliances. Also, MANETs are the suitable solutions in the emergency situations, for example when earthquakes and natural disasters have destroyed the existing infrastructure networks. One of the most important ad-hoc applications is the sensor networks. Sensor networks have been addressed by many researchers in recent years because they have a lot of potential applications. Akyildiz et al. [Akyildiz 2002] gave a clear description about sensor networks and their nature and applications.

The main characteristics of MANET are identified as follows [Li 2004]:

- Autonomous: Each node in MANET is autonomous and works as router and host.

• Distributed: MANET is distributed in its operation and functionalities, such as routing, host configuration and security. For instance, unlike wired network, MANET cannot have a centralized firewall.

• Multi-hop: If the source and destination of a message is out of the radio range of one node, a multihop routing is necessary.

• Dynamic topology: Nodes are mobile and can join or leave the network at any time; therefore, the topology is dynamic.

• Thin terminal: The mobile nodes are often light weight, with less powerful CPU, memory and power.

1.2 Attacks against Ad Hoc Networks

As MANETs become widely used, the security issue has become one of the primary concerns. For example, most of the routing protocols proposed for MANETs assume that every node in the network is cooperative and not malicious [Zhang 2003]. Therefore, only one compromised node can cause the failure of the entire network.

There are both passive and active attacks in MANETs. For passive attacks, packets containing secret information might be eavesdropped, which violates confidentiality. Active attacks, including injecting packets to invalid destinations into the network, deleting packets, modifying the contents of packets, and impersonating other nodes violate availability, integrity, authentication, and non-repudiation. Examples of ad hoc networks attacks can be found in [Chun2003],[Chun2-2003],[huang2004] and [Awerbuch 2004].

Proactive approaches such as cryptography and authentication, which considered as protection techniques, were first brought into consideration, and many techniques have been proposed and implemented. However, these mechanisms are not sufficient. If we have the ability to detect the attacker once he comes into the network, we can stop him before doing any significant damage to the network or any data. Here is where the intrusion detection system comes in [Xiao 2006]. IDSs (Intrusion Detection Systems) should complement existing prevention techniques, which considered as the first layer of defense, in order to improve the network performance and provide a highly survivable system [Sun 2004]. Many researches [Zhou 1999] [Papadimitratos 2002] [Hu 2002] [Hu2 2002] have been devoted to MANET prevention mechanisms, e.g. cryptography and authentication techniques, especially focused on the routing layer.

In this study, we focus on the dropping packets attacks. There are many reasons for dropping packets in ad hoc networks. We can classify these reasons into two main types: intended and unintended misbehavior. The unintended misbehavior could be caused by many causes such as node overloaded (due to lack of CPU cycles or limited buffer space), network congestion or collision. Because wireless channels are known to be unreliable [Tanapat 2008], packet dropping may be occurred due to link errors such as interference or fading.

On the other hand, for intended misbehavior, we call the node that causes this type of misbehavior as misbehaving node. Furthermore, we classify this type of intended misbehavior into two types: the first type is the selfish misbehavior where the node participates to carry the routing control packets (the packets of discovery and maintenance phases) to extract useful information from it. However, these types of nodes

do not participate to route the data packets in order to save its limited energy and network bandwidth. The second type of the intended misbehavior is the malicious behavior where we call the node that doing this type of misbehaving as a malicious node. As the selfish node, this node also participates in routing the routing packets but does not participate in routing the data packets. The main purpose of this attack is to disrupt the network and affect its connectivity or availability. Figure (1) describes this classification of dropping packets misbehavior.

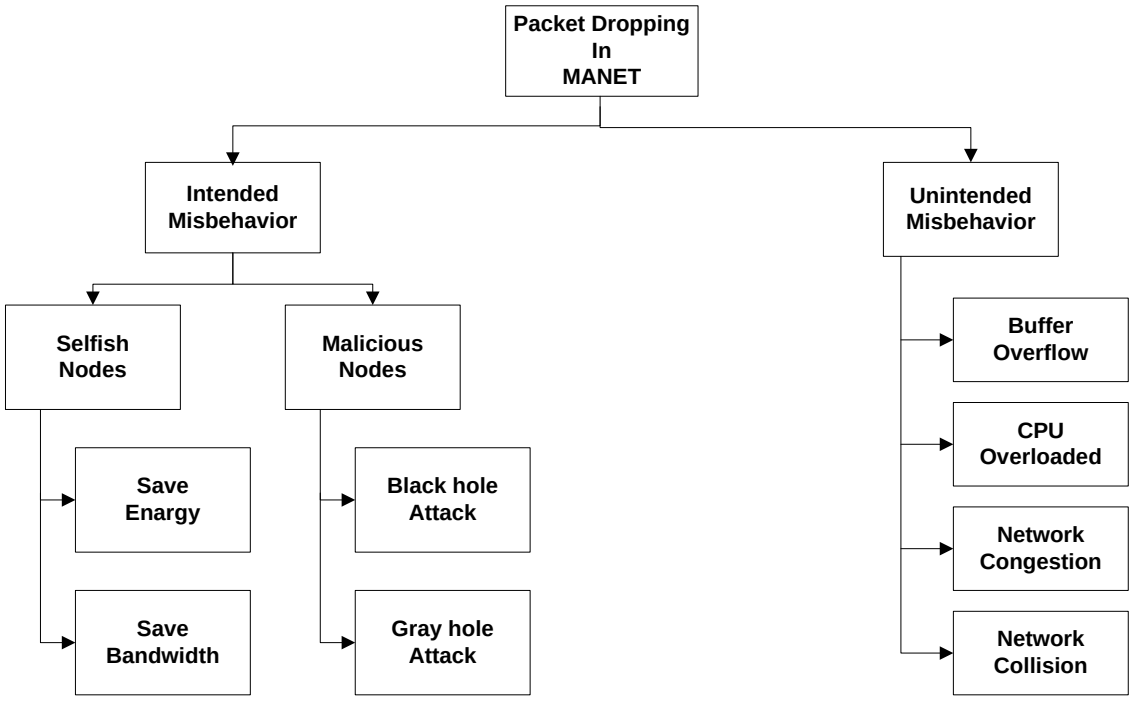


Figure 1-1: Packet dropping in MANET

1.3 MANET Routing Protocols

There are many proposed routing protocols to MANET. In this section we give a brief description about these types and we will focus on the DSR routing protocol, which we will use in our simulations.

1.3.1 MANET Routing Protocols Overview

Larsson and Hedman [Larsson 1998], and Sun in his PHD research [Sun 2004] have given a good overview on the existing routing protocols. Many routing protocols have been proposed for MANETs. In general, these protocols could be divided into three categories: proactive, reactive, and hybrid. In proactive category, the routing protocol find path to every node in the network whether there is a packet ready to be sent or not, and update these paths frequently after specific period of time. In contrast, the reactive routing protocols do a route discovery to find the destination of the packet, only if there is a packet ready to be sent, that is why we call it on demand routing protocols. Whereas, the hybrid routing protocols could act as a proactive routing protocol if the destination in the same area of the sender, and act as a reactive routing protocol if the destination is outside the area of the sender. Figure 1.2 illustrates the classification of these protocols and give examples for each type.

Proactive routing protocols, such as Destination-Sequenced Distance Vector routing protocol (DSDV) [Perkins 1994], the Wireless Routing Protocol (WRP) [Murthy 1996] and OSLR [Clausen2003], waste limited bandwidth by continuously maintain the complete routing information about the whole network. They react to topology changes, even if there is no traffic. They are also called table-driven methods. The protocols in this

category differ in the number of tables maintained and the information that each table contains as well as the details of how they are updated.

Reactive routing protocols (such as Ad-hoc On-demand Distance Vector routing protocol (AODV) [Perkins 1999], the Temporally Ordered Routing Algorithm (TORA) [Park 1997], and the Dynamic Source Routing protocol (DSR) [Johnson 2004]) are based on demand for data transmission. They could significantly reduce the routing overhead when the traffic is lightweight, since they do not need to periodically update route information and do not need to find and maintain the routes when there is no traffic. The differences among reactive routing protocols lie in the implementation of the path discovery mechanism and optimizations to it.

Hybrid methods combine proactive and reactive methods to find efficient routes. ZHLS [Joa-Ng 1999] is one example of hybrid routing protocols. In ZHLS, the whole network is divided into *nonoverlapping* zones. ZHLS is proactive if the traffic destination is within the same zone of the source. It is reactive because a location search is needed to find the zone ID of the destination. Also, Zone Routing Protocol, ZRP, a hybrid routing protocol suitable for a wide variety of mobile ad-hoc networks, especially those with large network spans and diverse mobility patterns [Sygmunt2003].

In this thesis, we use DSR as the routing protocol since our work needs a source routing protocol, to know the whole nodes in the path before sending a data packet. Because of that the DSR routing protocol will be described in more details in the next section.

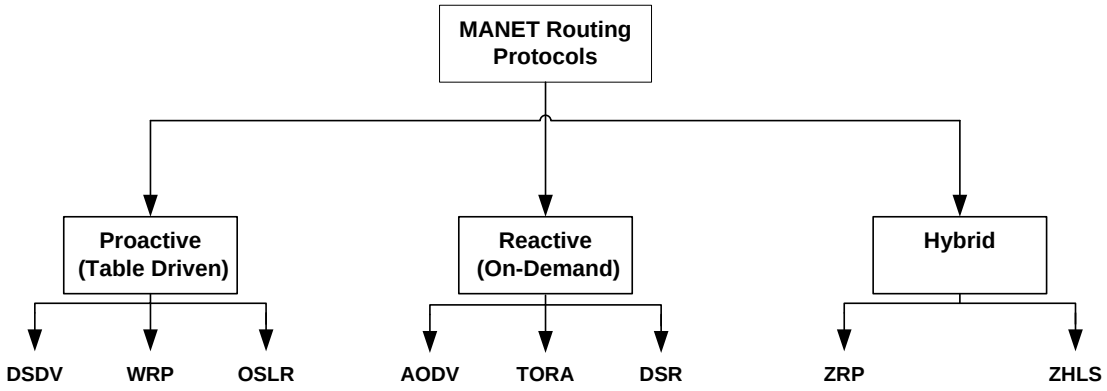


Figure 1-2: MANET routing protocols classification

1.3.2 DSR Basic Operations

In this research, the AACK IDS system is built over DSR [Johnson 2004] routing protocol because, as we have discussed in last section, our mechanism need a source rout protocol and also Watchdog and TWOACK techniques, which we will study in this research, used it. Furthermore, Watchdog is restricted to work just under DSR routing protocol because it needs to know the entire path that the packet will use to reach its destination [Marti2000].

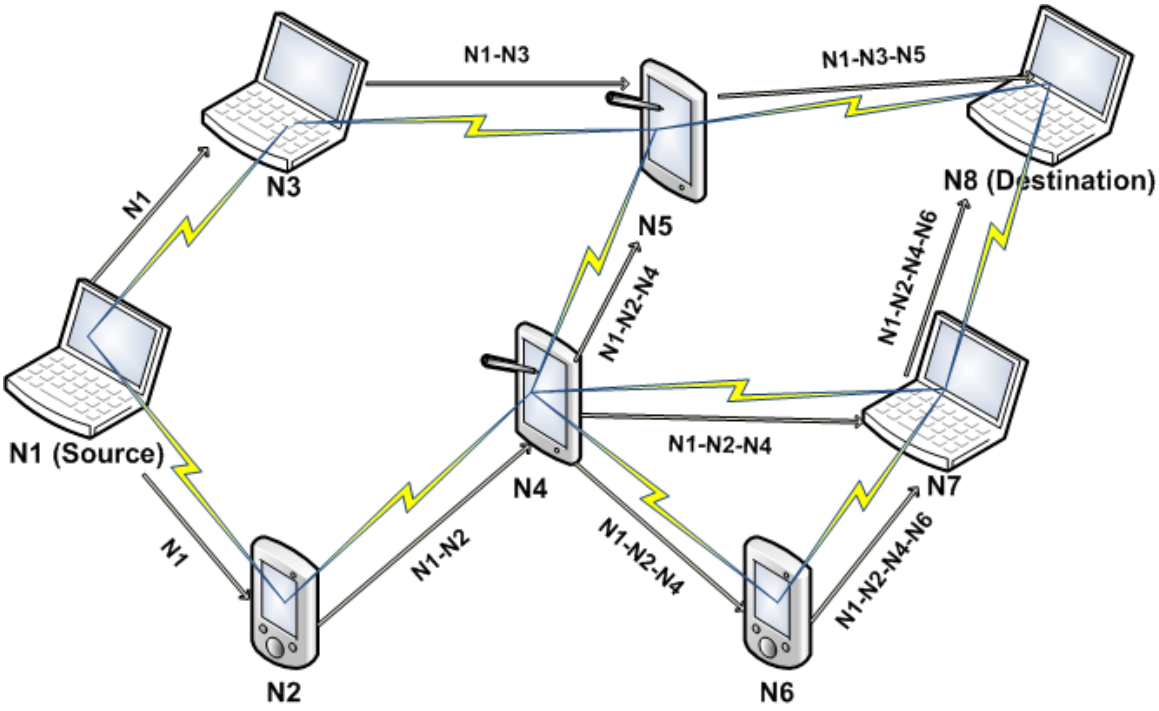
When a node has a ready packet to send, it searches its cache to find a rout to the packet's destination. If there is no route to the destination in the cache then the sender will broadcast a Route REQuest (RREQ) message (illustrated in figure 1.3) to its neighbors to ask for route to the packet destination. Each node that receives the RREQ from the sender will append its address to the source route in the packet header and rebroadcast the RREQ to its neighbors. If any node receives the same RREQ another time it will not resend it again and just ignore it. When the RREQ reaches the destination node, it will create a Rout REPly (RREP) send it (as unicast packet) back to the initiator

of the RREQ (the sender) by reversing the route that it extracts it form the source route of the RREQ packet that it receives. Figure l.3.1b, describes this process which represents the *route discovery* phase of DSR. The second phase is called *maintenance phase*; in this phase the node generates a Route ERRor massage to inform the source if there is a link breakage.

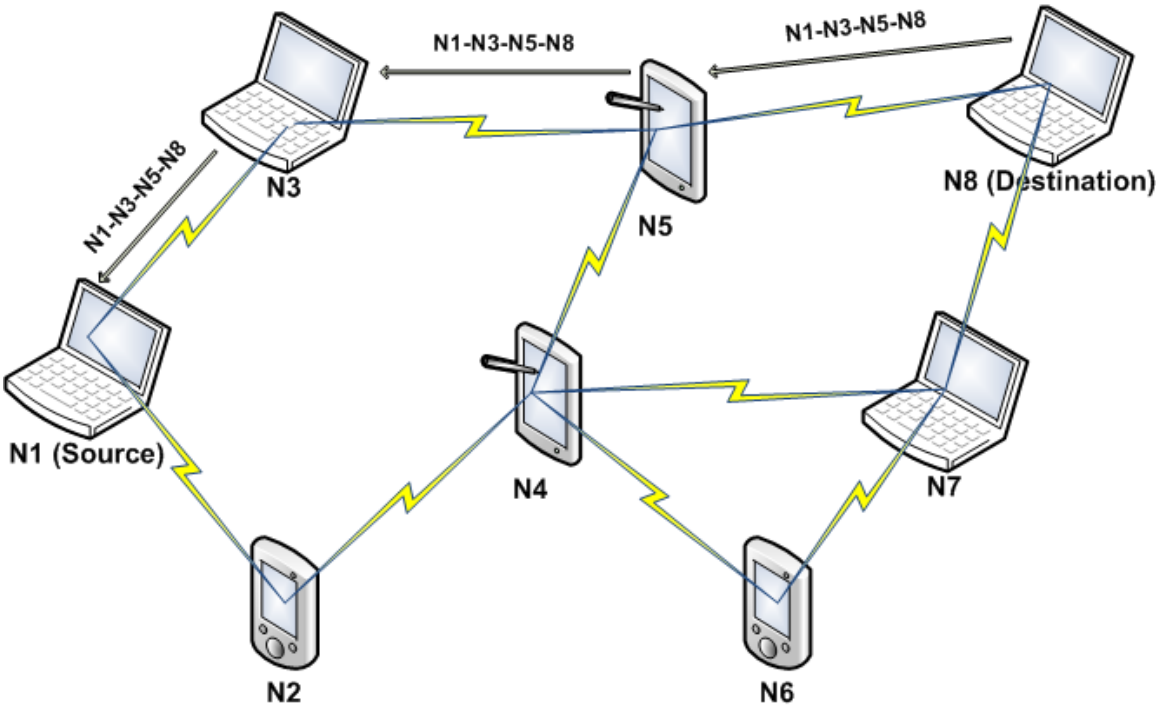
1.4 Intrusion Detection System (IDS) Overview

Intrusion detection can be defined as a process of monitoring activities in a system, which can be a computer or network system. The mechanism by which this is achieved is called an intrusion detection system (IDS). An IDS collects activity information and then analyzes it to determine whether there are any activities that violate the security rules. Once an IDS determines that an unusual activity or an activity that is known to be an attack occurs, it then generates an alarm to alert the security administrator. In addition, IDS can also initiate a proper response to the malicious activity. Although there are several intrusion detection techniques developed for wired networks today, they are not suitable for wireless networks due to the differences in their characteristics. Therefore, those techniques must be modified or new techniques must be developed to make intrusion detection work effectively in MANETs. [Xiao 2006]

Intrusion prevention measures like encryption and authentication can only prevent external nodes from disrupting traffic, but can do little when compromised nodes internal to the network begin to disrupt traffic [Mishra 2004]. Many historical events have shown that intrusion prevention techniques alone which are usually a first line of defense are not sufficient. As the system become more complex, there are also more weaknesses, which lead to more security problems.



(a) Route Request



(b) Route Reply

Figure 1-3: Route request and route reply in DSR

Unlike firewalls which are the first line of defense, intrusion detection can be used as a second wall of defense, to protect the network from such problems, and comes after intrusion has happened and a node has been compromised. If the intrusion is detected, a response can be initiated to prevent or minimize damage to the system.

Many researchers [Brutch 2003][Mishra 2004][Xiao 2006] have classified the existing IDSs as either host-based or network-based, depending on the data collection mechanism. Host-based IDS operate on the operating system's audit trails, system and application logs, or audit data generated by loadable-kernel modules that intercept system calls. Network-based IDS operate on packets captured from network traffic. In addition, the IDSs may be classified based on the detection technique as described below:

- **Signature-based detection systems:** The system keeps signatures of known attacks and uses them to compare with the captured data. Any matched pattern is treated as an intrusion. This technique may achieve low false positive rates, but does not perform well at detecting previously unknown attacks. Like a virus detection system, it cannot detect new kinds of viruses.
- **Anomaly-based detection systems:** The normal profiles (behaviors) of users are kept in the system. The system compares the captured data with these profiles, and then deal with any activity that deviates from the baseline as a possible intrusion by informing system administrators or initializing a proper response. This system is suitable for unknown attacks but it gives high false positives rates.
- **Specification-based detection systems:** The system defines a set of constraints that describe the correct operation of a program or protocol. Then, it monitors the execution of the program with respect to the defined constraints. This technique may

provide the capability to detect previously unknown attacks, while exhibiting a low false positive rate.

Another classification based on IDS architecture classifies the existing IDSs to three categories: stand alone, distributed and hierarchical. A more detailed taxonomy and information about IDSs can be found in [Xiao 2006] [Axelsson 2003] [Debar 2000] [Mukherjee 1994]**.**

1.5 Research Overview

In this section we give a background about this research and explain its challenges. In addition, we describe the motivation and research main contributions in the last subsection of this section.

1.5.1 Research background

The early intrusion detection system for MANETs was developed by Marti et al. [Marti 2000], which is called ***Watchdog*** technique. Where each node will monitor the next hop (overhearing using promiscuous mode) to assure that it will forward the packet or not. As described in the same research, it has several weaknesses, i.e., ambiguous collision, receiver collisions, limited transmission power, false misbehaving, collusion, and partial dropping. These weaknesses will be described in the next chapter (literature review). There are many researches that are dedicated to solve these problems.

Actually, our concern was to solve the receiver collision and limited transmission power. Where in these vulnerabilities the misbehaving node can cheat the sender (monitor) of the packet by make it believe that it correctly forward the packet whereas it does not. One solution for those problems was proposed in 2005, which is the ***TWOACK***

scheme [Balakrishnan 2005]. It is an acknowledgment based scheme that assures transmission of the packet over three consecutive nodes; thereby, the overhearing will not used anymore. This mechanism solve the two problems but with significant overhead and more calculations that will affect the power and memory of the network, which are scarce resource for MANETs. Furthermore, this technique is detecting the misbehaving links rather than misbehaving nodes, which is considered as a weakness because in such way the misbehaving node will get more chances to drop data packets.

Our proposed mechanism, AACK will be described in more details in chapter 3, concerns were to improve the TWOACK weaknesses, which are, the routing overhead and detecting misbehaving links rather than nodes. Therefore, we consider our proposed solution for the watchdog two problems that we have mentioned as an enhancement to the TWOACK scheme.

1.5.2 Research Challenges

It is very challenging to design an intrusion detection system for mobile ad-hoc networks. The lack of fixed infrastructures and administration points make it difficult to collect audit data for the entire network. Furthermore, we have to consider the scarce MANET resources (such as limited wireless bandwidth, computation ability and energy supply, etc.) when we design the IDS system for MANETs. Also, mobility makes the distinction between false alarms and real intrusions very difficult. For example, a node that sends wrong routing information could be because it has been compromised, or because of its arbitrary movement. Reducing false alarms while increasing the network throughput and minimizing the routing over head are very challenging tradeoffs. Those

tradeoffs will be clear when we adjust the important two parameters of the IDS system which are timeout and dropping threshold.

1.5.3 Research Contributions

The following is a list of contributions of this thesis:

- Studied the effect of packet dropping misbehaving on mobile ad hoc networks using simulations.
- Proposed a new IDS for MANETs, which solve the two problems of watchdog technique, receiver collision and limited power transmission, improves the performance of existing mechanisms (TWOACK and Watchdog).
- Compared intrusion detection mechanisms in various wireless scenarios
- Examining video traffic over intrusion detection systems over MANETs

1.6 Organization

This thesis is organized as follows. Chapter 1 gives an overview and background of wireless ad hoc networks, intrusion detection systems, and thesis. Chapter 2 presents the literature review and problem statement of this research. Chapter 3 describes a system models and implementation for detection mechanisms, namely watchdog, TWOACK, and AACK. Chapter 4 presents simulation environment, studies the detection mechanisms in various scenarios, and discusses the results. Lastly, conclusions, limitations, and future work are described in chapter 5.

2 PROBLEM STATEMENT AND LITERATURE REVIEW

In this chapter we describe our problem statement in the context of literature review of watchdog mechanism. It consists of three main sections: the first one is the problem statement section, where we describe the watchdog technique with more details in its problems. The second section explores the related work in field of intrusion detection systems in MANET. Limitations of current intrusion detection systems are discussed in section three.

2.1 Problem Statement

This research mainly starts to solve two problems of watchdog intrusion detection system. In this section we describe watchdog IDS in more details. At the end of this section, we identify our research problems.

2.1.1 Watchdog Mechanism

This research basically considered as an improvement to the watchdog mechanism. Therefore, we will describe it in more details in this section, especially its weaknesses.

Watchdog is the base intrusion detection technique that many of the recent researches depended on. It was proposed by Sergio Marti et al. [Marti 2000], they proposed two techniques that improve throughput in MANETs in the presence of misbehaving nodes that agree to forward data packets but rather they drop all data packets. Marti et al. classify the reasons that make node to misbehave. Node may misbehave because it is overloaded, selfish, malicious, or broken. An overloaded node

lacks the CPU cycles, buffer space, or available network bandwidth to forward packets. A selfish node is unwilling to spend battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to it, even though it expects others to forward packets on its behalf. A malicious node launches a DoS (Denial of Service) attack by dropping packets. A broken node might have a software fault that prevents it from forwarding packets.

They used two techniques to mitigate the decrease in the throughput due to the above node categories, the first one is the intrusion detection which is the Watchdog that identify misbehaving nodes and the other is the response of the intrusion detection system which is a pathrater that helps routing protocols avoid these nodes. When a node forwards a packet, the node's Watchdog verifies that the next node in the path also forwards the packet. The Watchdog does this by listening promiscuously to the next node's transmissions. If the next node does not forward the packet, it is misbehaving. Every time a node fails to forward the packet, the Watchdog increments the failures counter. If the counter exceeds a certain threshold, it determines that the node is misbehaving; this node is then avoided using the pathrater. The pathrater, run by each node in the network, combines knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable, it uses the reliability metric instead of shortest path. Each node maintains a rating for every other node it knows about in the network. It calculates a path metric by averaging the node ratings in the path. The Watchdog technique has its own advantages and weaknesses. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of:

- Ambiguous collisions: These prevent node A from overhearing the transmission from node B, as shown in Fig. 1.
- Receiver collisions: Node A can only tell whether B has sent a packet, but not if node C received it or not, as shown in Fig. 2.
- Limited transmission power: A misbehaving node could limit its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the true recipient. As shown in Fig. 3.
- False misbehavior: This occurs when a node falsely reports other nodes as misbehaving.
- Partial dropping: A node can circumvent the Watchdog by dropping packets at a lower rate than the Watchdog's configured minimum misbehaving threshold.

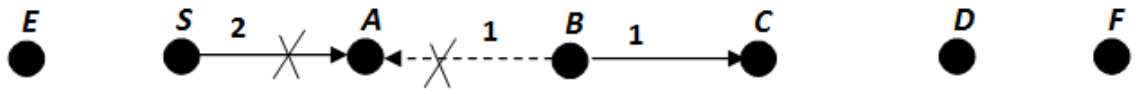


Figure 2-1: Node A does not hear node B forward packet 1 to C, because B's transmission collides at A with packet 2 from source S.

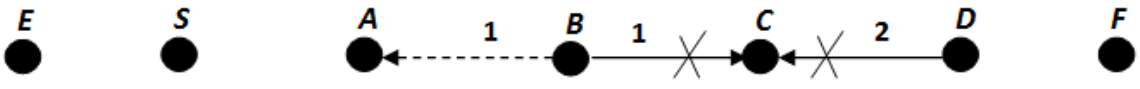


Figure 2-2: Node A believes that node B has forwarded packet 1 to C, although C never received the packet due to a collision with packet2.

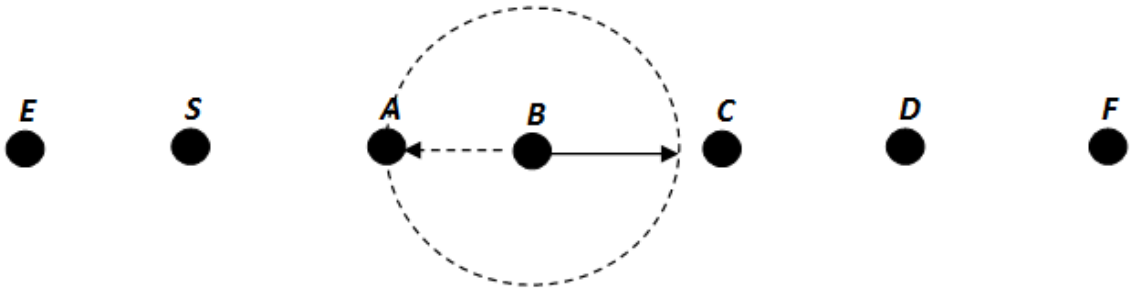


Figure 2-3: Node B limits its transmission power such that the signal is strong enough to be overheard by node A but too weak to be received by node C.

2.1.2 Research Problem Statement

Because there are many intrusion detection systems are built based on watchdog technique, we focus or research to solve some of the watchdog problems. As we will describe in the related work, next section, some of the Watchdog problems, which have been mentioned above, are discussed and solutions are proposed for those problems. In this thesis we proposed a solution to tackle two problems of watchdog, which are the receiver collisions in figure 2 and the limited transmission power in figure 3.

2.2 Related Work

The IDSs is considered as a second layer of defense, it should be a complement for existing prevention techniques [Sun 2004]. There are many researches that have been devoted to improve protection of MANETs against misbehaving nodes [Zhou1999][Hu2002][Hu2-2002][Zapata2002][Patwardhan2005]. Actually, our research is focused on intrusion detection techniques. There are several classifications for the proposed intrusion detection systems. As discussed in chapter 1, Ping [Yi 2005] classified the existing IDSs based on the detection algorithm to anomaly, misuse, specification-based detection. Also, [Tanapat 2008] classified them into reputation-based and

incentive-based IDS, and this is what we follow in our literature review as shown in figure 2.4.

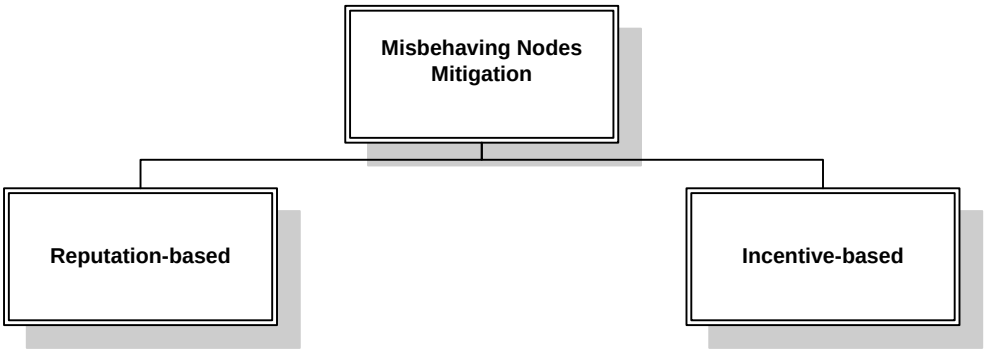


Figure 2-4: MANETs IDS Classification

Before we start to explore the IDS mechanisms, a general architecture for IDS in MANET is presented by Yougguang et al. [Zhang 2000]. It gives a general view how IDS is implemented, in most cases, over MANETs. They proposed general intrusion detection and intrusion response architecture for MANETs. An individual IDS agent is attached to each mobile node, and each node in the network is responsible for both intrusion detection and intrusion response. Collaborative decision making: Each node participates actively in the intrusion detection process. Once one node detects an intrusion with confidence high enough, this node can start a response to the intrusion. In a simple implementation of this design as shown in figure 2.5, a majority voting scheme is used to determine whether attack happens.

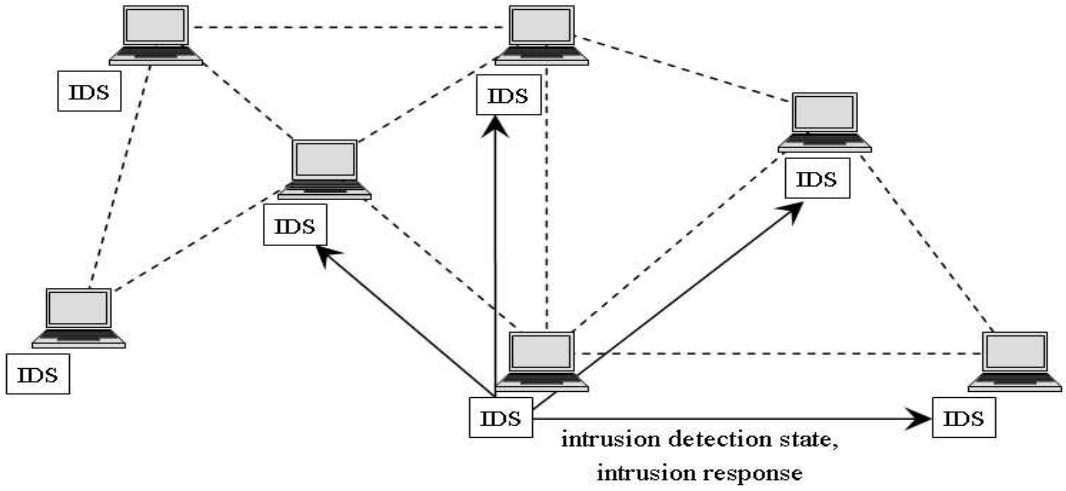


Figure 2-5: General intrusion detection and response system

2.2.1 Reputation-based mechanisms

A reputation-based mechanism uses a reputation system in order to detect and rate misbehaving nodes. A clear definition for reputation system is presented by [Buchegger2005] as the performance of a node participating in the base protocol as seen by others. Although the watchdog is considered as a reputation-based mechanism, we described it in the previous section because it is highly related to our problem statement.

CORE, a ***CO***llaborative ***RE***putation mechanism proposed by Michiardi et al. [Michiardi 2002], also use a *watchdog* mechanism; however it is complemented by a sophisticated reputation mechanism that differentiates between subjective reputation ,which is gained by observations, indirect reputation that uses positive reports by others, and functional reputation. These three components are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Each node participates in the IDS and has reputation table and watchdog mechanism. The reputation table keeps track of reputation values of other nodes in the network. Since a misbehaving node can send accuse a good node, just a positive rating

factors can be distributed in CORE. A performance analysis by simulation is stated for future work. This mechanism still uses the watchdog mechanism with its disadvantages and problems.

Sonja Buchegger and Jean Boudec proposed another reputation mechanism that is called “CONFIDANT", which stands for ***C***ooperation ***O***f ***N***odes: ***F***airness ***I***n ***D***ynamic ***A***d-hoc ***N***e***T***works [Buchegger 2002]. CONFIDANT has four main components, namely a monitor, a reputation system, a path manager, and a trust manager. These components are required to be implemented in every node. Each node monitors its neighbors by listening to the transmission of the next node or by watching routing protocol behavior. A trust manager is used to manage ALARM messages, which are sent when a misbehaving node is detected. The reputation system is used to rate every node in a network. A path manager is responsible to rank a path according to a security metric, e.g., reputation of the node in the path and to get rid of any path containing a selfish node. In addition, a path manager will penalize a selfish node by denying all services to it. Through a study the protocol performance, the authors showed that the throughput given by CONFIDANT in a scenario when a third of nodes behave selfishly is very close to the throughput of a normal network condition without selfish nodes. However, it is also still depends on the watchdog mechanism which still has many problems.

“CineMA", stands for ***C***ooperat***i***o***n*** ***E***nhancement in ***MA***NETs, is proposed to respond to a misbehaving node by limiting the number of packets forwarded by it [Frank 2004]. Likewise CORE and CONFIDANT, CineMA depends in watchdog technique and uses the same penalty scheme as in CORE and CONFIDANT by excluding the misbehaving node from the routing phase. Unlike CORE and CONFIDANT, not all the

nodes work IDS; CineMA only needs a group of nodes to perform necessary functions. It consists of three main modules including a *Watchdog* module, a reputation system module, and an interface queue module. A Watchdog module performs system monitoring to collect information. A reputation system uses collected information to determine the level of cooperation based on the number of received packets and the number of forwarded packets. These values are also used, at the interface queue module, to limit the amount of packets which a selfish node is allowed to transmit. CineMA requires the use of a cryptographic mechanism to ensure secure communications among all nodes implementing CineMA functions. Although, overall throughput and performance of CineMA have not been proven, a major advantage of CineMA is that it can limit the sending rate of a misbehaving node. That can mitigate the effect of false alarms by avoiding excluding the node completely.

Animesh and Amitabh [Patcha 2003] proposed another research that made an extension to the Watchdog technique as an improvement by tackling the problem of collusion attack, where more than one node collaborate to do a malicious behavior. Furthermore, they assumed that the few nodes that formed the network are trusted nodes and the others that would join the network later are ordinary nodes. The Watchdog nodes are selected from the trusted nodes, and thus they avoid the problem of false reporting. In every Watchdog, two thresholds are maintained for all its neighbors that are not trusted nodes. The first one is called the SUSPECT_THRESHOLD, a measure of node's misbehaving, and the other is called ACCEPTANCE_THRESHOLD, a measure of node's good behavior. And based on these thresholds the Watchdog node will identify the neighboring nodes as a malicious or trusted node. This mechanism is built over AODV

and aims to detect a black hole attack, dropping all data packets. Likewise the previous mechanisms, this mechanism also depends on watchdog technique. In contrast with our proposed mechanism, it is proposed to solve a collusion problem of watchdog in which two malicious nodes cooperate to do a malicious activity.

Parker et al [Parker 2004] improves and enhances the Watchdog technique, which is viable just for DSR routing protocol. Their proposed solution is applicable to all of the routing protocols used in MANETs, not just DSR. In contrast to the Watchdog, the nodes overhearing all the other nodes in their proximity not just the next forward node on the path. They proposed two response mechanisms. The passive response mode where each node acts independently and eventually the intrusive node will be blocked from using all network resources. The other mechanism is the active response mode where the decision making is done by a cluster head by initiates a voting procedure. If the majority determines that the suspected node is in fact intrusive, an alert will be broadcast throughout the network and the intrusive node will be blocked from using network resources.

One solution that solves the problems of the receiver collision and limited transmission power is proposed by Balakrishnan et al [Balakrishnan 2005]; it is an acknowledgment-base protocol, which is called TWOACK. It does not rely on overhearing other nodes in its vicinity; thereby, in contrast to all the previous mechanisms, it does not use a watchdog technique. This scheme can be added on to a source routing protocol such as DSR. It acknowledges every data packet transmitted over every three consecutive nodes along the path form source to destination. Suppose node A has discovered a route to F with a source route A → B → C → D → E → F. In

TWOACK when B forwards a packet for A, C (the node two hops away from A) receives the packet and sends an acknowledgement to A indicating B has forwarded the packet properly. If A does not get an acknowledgement for the packet, it expected to be forwarded by B to C, within a certain timeout period it suspects B to be misbehaving. The same procedure is carried out by every set of three consecutive nodes along the source route. In TWOACK each forwarded packet has to be acknowledged which may contribute to traffic congestion on the routing path. They also proposed S-TWOACK (Selective TWOACK) to reduce this extra traffic by sending a single acknowledgement for a number of packets instead of for a single packet.

As they mentioned in their paper that S-TWOACK reduces the overhead but with cost of less packet delivery ratio. That is because the malicious node will take more time to drop more packets. Even with the S-TWOACK, this scheme still adds more overhead to the routing protocol because of the many acknowledgments for each packet along the path especially in cases of overload and long paths. Furthermore, it has less detection accuracy because it detects links rather than nodes. Thus, the malicious node will has several chances even if it will be detected in one link it can works on other link. Because our proposed mechanism is considered as improvement for TWOACK, this mechanism will be discussed in more details in chapter 3.

Islam et al. [Islam 2005] have proposed another way to solve the problems of receiver collision and limited transmission. For a flow f each node, h hops away from the source in the routing path, measures the rate R[f, h] at which it processes packets. At the source, intermediate and destination nodes processing of packets refers to sending, forwarding and receiving packets respectively. The value of h for source is 0 and it

increases for subsequent nodes towards the destination. At the end of each period T, the destination puts R[f, h=destination] in a packet called the route status packet (RSP) and sends it to the source through all the intermediate nodes of f. Each intermediate node also appends R[f, h] to RSP before sending to the next node. R[f, h] can be digitally signed by its respective node to prevent other nodes from modifying it. When RSP reaches the source node, it contains R[f, h] values of all the downstream2 nodes of f. Now we can estimate the forwarding ratio of a node h hops away from the source by the following expression:

Forwarding ratio, $F[f, h] = R[f, h + 1] / R[f, h - 1]$

If Delivery ratio, $R[f, h = \text{destination}] / R[f, 0] < Rthres[f]$

Where Rthres[f] is the allowable minimum end-to-end delivery ratio for the flow f, the source suspects the intermediate node, h hops away from the source with the highest F[f,h], is dropping packets at an intolerable rate. The source maintains a misbehavior counter MC[f,h] for each downstream node. If MC[f,h] reaches a threshold for a downstream node, the source declares the node to be misbehaving. This approach minimizes the overhead on the network traffic that was in the approach of Balakrishnan et al. but here the IDS system depends on information that comes from the misbehaving nodes. The adversary node can change its R[f,h] value which could lead to increase false alarms.

In [Hasswa 2005], Hasswa et al proposed a novel intrusion detection and response system called Routeguard. In this technique the two techniques that were proposed by Marti et al., Watchdog and pathrater, are combined to classify each neighbor node as:

fresh, member, unstable, suspect, or malicious. The class of each node depends on the ratings obtained from the Watchdog according to its behavior. Furthermore, each class has a different trust level which goes from member (trusted), which allows the node to participate in the network, to Malicious (untrusted), which is completely untrusted and gets banned from the network. Routeguard is run by each node in the network and stores a rating for all the nodes it knows, and this is similar to the process of pathrater. Routeguard improves Pathrater by assigns ratings to nodes and calculates a path metric in a refined way. Furthermore, Routeguard introduces a more detailed and natural classification system that rates each node in the network. However, it is still using a watchdog technique with its all problems.

Nasser and Chen [Nasser 2007] proposed IDS that is considered as an extension for watchdog by solving a problem of collusion attack. In this work, they proposed an enhanced intrusion detection system for discovering malicious nodes in MANETs called ExWatchdog. ExWatchdog extends the Watchdog proposed by Marti et al. In this paper they solve one of the problems that we have introduced above in the weaknesses of the Watchdog technique which is the false misbehaving problem, a malicious node falsely reports other nodes as misbehaving while in fact it is the real intruder. A table is maintained by each node, this table records the number of packets the node sends, forwards or receives respectively. When the source receives a report about misbehaving node, he will find another path to ask the destination node about the number of received packets. If the number of received packets equal to the number of packets that the source has sent, then the real malicious node is the node that reports others nodes as misbehaving. Otherwise, nodes being reported malicious do misbehave. However, there

is a limitation in this technique. If the true misbehaving node is in the all available paths from source to destination, as shown in figure 2.6, then it is impossible to confirm and check the number of packets with the destination. Furthermore, it conditions that there are another path to the destination.

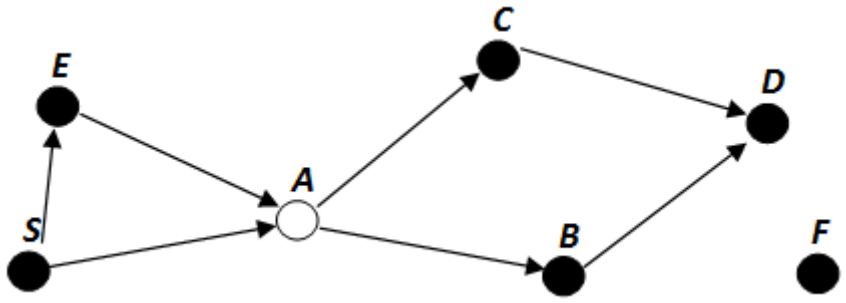


Figure 2-6: A is a malicious node that falsely reports all nodes on the available paths from source to destination as misbehaving in order to affect the availability of the network.

Like all the previous proposed mechanisms, Cop mechanism [Tanapat 2008] is proposed to improve the network performance in which misbehaving nodes are presented. Tanapat in his dissertation proposed a Cop mechanism to minimize the power consumption in watchdog mechanism by assigning the monitoring and detection functions to some of the nodes rather than all nodes. Cop nodes are selected as trusted nodes, and they are deployed by static placement to cover the whole area or the moves around the whole area of the network and do the monitoring for misbehaving nodes. Cops improves watchdog in terms of power consumption but still use watchdog and by that it suffering the same problems of watchdog that we propose a solution for some of it. Furthermore, in case of mobile cops there would be some of the regions that stays without monitoring whereas in case of all nodes applying detection functions this will not be happened.

Table 2-1: Reputation-based IDSs summary

Mechanism	Published Date	RP	Detection Function	Misbehaving Detected	Use WD	Problems Solved
Watchdog	2000	DSR	All nodes	**A**ll **P**acket **D**rop (APD)	Yes	None
CORE	2002	All	All	**S**elective **P**acket **D**rop (SPD)	Yes	Partial dropping
CONFIDANT	2002	DSR	All	APD + Routing Attacks	Yes	None
Patcha	2003	AODV	Some	APD	Yes	Collusion
CineMA	2004	DSR	Some	SPD	Yes	Partial dropping
Parker	2004	All	Some	APD	Yes	None
TWOACK	2005	DSR	All	APD	No	RC+TC
Routeguard	2005	DSR	All	SPD	Yes	Partial dropping
ExWatchdog	2007	DSR	All	APD	Yes	False Misbehaving
Cop	2008	DSR	Some	APD	Yes	None

2.2.2 Incentive-based mechanisms

The incentive-based mechanisms do not be considered as detection systems because they just motivate the network nodes to cooperate with each other in forwarding packets. However, we prefer to select some of this type of researches because it is considered as misbehaving nodes mitigation.

In this type of mechanisms the nodes have to pay to forward their packets. To do that, these mechanisms use virtual money (or credits) [Buttyan 2003] or apply incentive mechanism. Any node that participates in packet forwarding will gain credits and it will use these credits to forward its own packets. If the node does not participate in

forwarding the packets of other nodes, it will not collect credits to forward its own packets.

In [Buttyan 2003] Buttyan and Hubaux proposed the use of virtual currency, called nuglets. When a node forwards a packet for others, a nuglet counter of that node is incremented by one. When a source node wants to send a packet, it must have enough nuglets, which is more than the number of intermediate nodes required to reach a destination. If a source node has enough nuglets, it can send packets. Otherwise, it must first collect more nuglets by forwarding packets from other source nodes. A tamper proof device must be used to maintain the nuglet counter. Thus, it depends on a hardware part to be established on each node, which makes it more difficult to be implemented in real scenarios.

In [Zhang 2004], Zhang proposed a Secure Incentive Protocol (SIP) that uses a session-based approach rather than per-packet-based approach. However, SIP assumed the use of a tamper-proof module, similar to the work in [Buttyan 2003]. In the SIP protocol, a session initiator and a session responder (a source-destination pair) will be charged for a service and intermediate nodes are rewarded with credits when they forward packets for a source-destination pair. SIP consists of three phases, i.e., Session initialization phase, Data forwarding phase and Rewarding phase. Each intermediate node is awarded a number of credits based on the number of forwarded packets.

Most of the incentive-based approaches use a security hardware part to hold the counter of the credits. This is improper choose to be applied in limited resource devises like ad hoc network devices.

2.3 Limitation of current IDSs

To mitigate packet dropping in mobile ad hoc networks, many solutions have been proposed. As we have discussed in the previous section, these mechanisms are classified into incentive-based, which is not applicable yet [Tanapat 2008], and the reputation-based. Our concentration was on the reputation-base intrusion detections. Most of these mechanisms, as we have discussed in the previous section, depends on watchdog technique. Some of them have focused on improving the detection efficiency without solving the problems that are stated by the watchdog mechanism paper [Marti 2000]. And the others focus on extending watchdog mechanism to work with other MANET routing protocols rather than DSR; or to detect selective packet dropping rather that all data packet dropping attacks. Some of them extended watchdog to tackle another type of attack, e.g. routing attacks, like in CONFIDANT [Buchegger 2002].

The only one that is considered as replacement to watchdog is the TWOACK mechanism [Balakrishnan 2005]. It tackles the two problems that we considered in our research. However, its detection system is not effective because it detects the misbehaving on link level rather than node level. That means it detects misbehaving links instead of nodes in which it gives more chance to the malicious node to drop more packets in each link it appears. Furthermore, it has more control packets overhead due to the TWOACK packets. These overhead packets will increase the network errors, e.g. network congestion and network collisions, which degrade the network performance. In addition, in TWOACK technique the intermediate nodes will do the detection functionality all the time, which will increase the computation time and use more

memory space due to the data base of sent packets; thereby, this will increase the power consumption, which affects one of the scarce resources of MANET nodes.

In AACK mechanism, minimizing TWOACK overhead while improving network performance is one of our considerations. In addition, we considered minimizing the power consumption by making the nodes that does detection functionality not working all the time. Instead, it works just if there will be a misbehaving activity along the path form source to destination. Finally, we improved the detection accuracy of TWOACK by introducing an algorithm to detect malicious nodes rather than links.

3 SYSTEM MODELS AND DESIGN

In this chapter, we discuss the mechanisms and models that we have studied in our work in more details. Basically, we implemented three mechanisms in our study. The first one is the early solution of the packet dropping attacks in MANETs which is the *watchdog* technique. The second is the *TWOACK* technique which is proposed to solve some of the *watchdog* problems and we will use it in our proposed solution with some enhancements. Finally, we discuss our proposed intrusion detection system, which is named *AACK*, and this phrase stands for ***A***daptive ***ACK***nowledgment IDS for MANT.

3.1 Overview

As we have discussed in the literature review, the malicious or selfish node can degrade the performance of ad hoc networks significantly by dropping the data packets silently. Throughout this chapter, we use the malicious node phrase to represent both selfish and malicious node because they have the same misbehavior, where both of them intend to drop data packets. Actually as we have described early, there are two types of packet dropping attacks. The first one, which we take in our consideration in this study, the malicious node will drop all the data packets while it participates in forwarding the routing packets this type is known as black hole attack. Similar to the first type is the second type attack, which is known as gray hole attack; the only difference is that the second type is smarter because it does not drop all the data packets. Rather, it tries to adjust its packet dropping rate to the detection threshold; thereby it makes it very difficult for the IDS system to detect such attack. However, the fist type (dropping all data

packets) effect on network performance is more than the second type because it is disrupt the network and affect its availability and connectivity. Furthermore, the false alarms are more in the case of second type. In this study, when we use the expression of packet dropping attack, we mean the first type, which is the black hole attack. This type is one of the Denial of Service attacks (DoS).

A simple and early solution to avoid packet dropping attacks is to use a watchdog technique as proposed in [Marti 2000]. However, this technique has many problems, as discussed in the literature review, especially for smart attackers who can cheat the monitor node and intend to do a collision at the next hop node by sending the packet while the received node busy with other transmissions. Also, a smart attacker can control its transmission power to let the monitor overhears its transmission while the next hop node is out of its range. We refer to these two problems as receiver collision and limited power transmission respectively, as it is called by watchdog mechanism in [Marti 2000]. These two problems have been discussed in more details in the problem statement section.

One of the most recently published solutions for these problems was The TWOACK mechanism [Balakrishnan2005]. It applies an acknowledgment-based approach to the routing layer to verify the delivery of data packet over every three consecutive nodes (each two hops) throughout the path from source to destination. Actually, this mechanism solved the two problems that we have mentioned before, but with more overhead. Furthermore, in contrast to watchdog, TWOACK detects a

misbehaving links instead of detecting misbehaving nodes. In this case, the TWOACK will not completely detect misbehavior node and it is still operate in the network and can drop more packets. That makes *TWOACK* less accurate in detecting misbehaving nodes.

In this research, we proposed the *AACK*, Adaptive Acknowledgment, mechanism which is an enhancement of *TWOACK*. Unlike *TWOACK*, our mechanism does a complete detection for the misbehaving node rather than a partial detection in *TWOACK* where it detects a misbehaving links. Furthermore, it decreases the overhead of the *TWOACK* acknowledgments for paths which has more than 2 hops. After we add the enhancement of node detection to the *TWOACK* we call it *E-TWOACK* and use it to implement our mechanism (*AACK*). *AACK* is called adaptive because it is a hybrid mechanism where it is composed of end-to-end acknowledgment and *TWOACK* schemes (two hops acknowledgment).

3.2 Model Assumptions

In this section we outline our assumptions as follows:

- Our mechanism works on a source routing protocol, e.g. in our implementation we use the routing protocol of Dynamic Source Routing (DSR) [Johnson 2004].
- Throughout this study, we assume a bi-directional communication between every pair of nodes in the network. That means, if node N2 can receive a packet from node N1, node N1 also can receive a packet from N2. Such a symmetry communication is required in our model for sending *AACK* and *E-TWOACK* packets in the opposite direction.

- We also assume that there is no collusion between misbehaving nodes, and the misbehaving nodes are capable of doing the following tasks:
    - Dropping any data packet.
    - Participating in the routing discovery and maintenance.
    - Controlling its transmission power to circumvent the watchdog monitor.
    - Capable of doing a collision at the receiving node by overhearing when receiving node is transmitting.

3.3 Node and Switching Models

In our implementations of the three schemes (watchdog, TWOACK, and AACK) we used two types of mobile wireless nodes: regular and malicious. We did some modifications to these nodes to appropriate or work. The main functions, modifications, and behavior of these nodes will be described in this section. A switching scheme, between AACK and E-TWOACK, is described also in this section.

3.3.1 Regular Node Model

To implement the AACK mechanism, the regular node must be modified to work properly with AACK. According to network simulator NS2 that we used in our simulations [NS2], the regular nodes of mobile ad hoc networks can be classified into three types of nodes based on the events that can be occurred by nodes:

- Source Node, which is the source where the packets are generated.
- Forwarder Node, which is the intermediate node in the path from source to destination that receive and forward the packet to the next hop until it reaches to the final destination.
- Destination Node, which is the final destination of the packet.

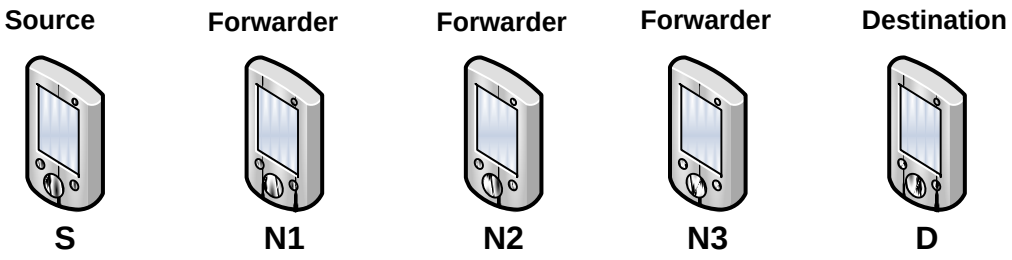


Figure 3-1: Nodes Classification

Figure 3.1 illustrates an example of the three types of nodes in the path. Where node S is the source node, D is the destination, and (N1, N2, N3) are the forwarder nodes. The functionality of the three types of nodes is modified to accommodate the AACK mechanism. Our modifications to these nodes are described as follows:

- ***Source Node***

We modified the source node to work in two modes and based on the node mode, the first mode is the AACK and the other is the E-TWOACK mode. To make source node capable of switching between the two modes, a switching scheme is developed. The switching algorithm will be described in the next section. For that purpose, we classified the data packets to two types: AACK (AA) packet, and TWOACK (TA) packet. When source node is in the mode of AACK it will send AA packets, while if it is in the E-TWOACK mode it will send TA packets.

- ***Forwarder Node***

The forwarder node, intermediate node, works based on the packet type. If it is a TWOACK packet (TA), the node will work according to the E-TWOACK scheme. For example, in the figure 3.1 after N3 received the TA packet it will send a TWOACK

acknowledgment to N1 and that is done for each three consecutive nodes throughout the whole path. In a different way, if the packet type is an AACK packet (AA), the node will just work as a regular node with basic DSR functions where it will forward the packet to the next hop.

The energy and memory consumption are important factors for MANETs. That is one of our scheme advantages over TWOACK. In TWOACK, the intermediate nodes do the calculations at the node all the time, whereas AACK works most of the time as a regular node with basic DSR functions. In such situation of mobile nodes that suffering from lack of power and memory spaces, our scheme will save more energy (no TWOACK calculations in CPU all the time of forwarding packets). In addition, because it will not save the AA packet id and its sending time, as it does in TA packets, it will save more memory space. Furthermore, the node will not send acknowledgment packets all the time of the session (sending all packets from one source to specific destination).

Figure 3.2 shows the procedure of forwarder node when it receives a TA and AA data packets, it is clear from the diagram that the forwarder node will normally forward that packet (without send TWOACK acknowledgment and register the id and forwarding time of the packet) in two cases, the first case when the packet type is AA (AACK). The other case, when the forwarder node is away from the source by just one hop.

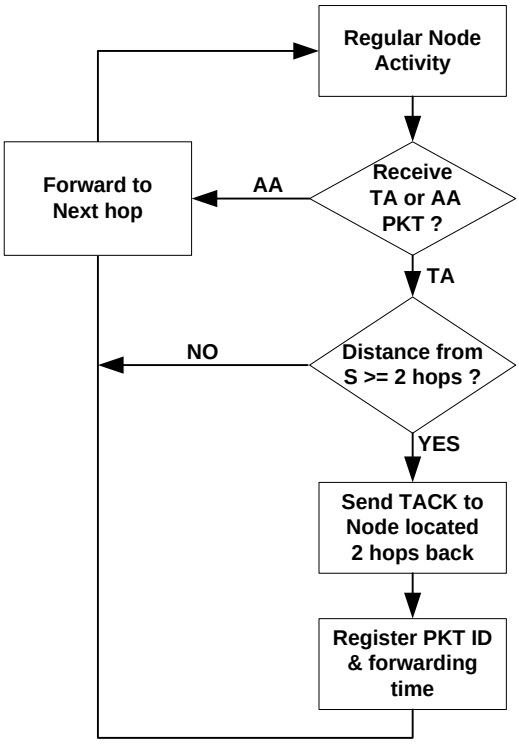


Figure 3-2: Forwarder Node Activity When receiving TA & AA data packets

- ***Destination node***

The destination node will also respond to the received data packet based on the type of the packet. If the received packet type is TA packet, the node will behave as the forwarder node when it receives TA data packet. Except that it will not register the packet id and the forwarding time because it is the final destination of the packet. In addition, it will also send a small packet switch to the source node to tell it to change its mode, this will be describe in more details in the next section of switching model. On the other hand, if the received packet type is an AA packet it will just send an AACK acknowledgment (end-to-end acknowledgment) to the source node, the originator of the packet. Figure 3.3 give a clear diagram of destination node response for received data packet.

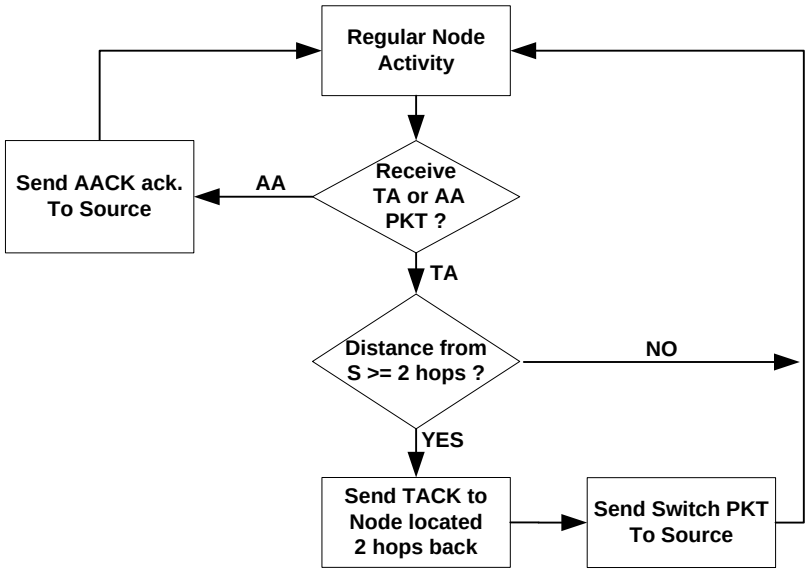


Figure 3-3: Destination node activity when receiving TA & AA data packets

3.3.2 Switching Model

This section will describe in details the switching model that we developed to work with our AACK scheme. As we have described in the last section, the system works in two modes (AACK and E-TWOACK) according to the state of the network (whether there are malicious nodes or not). That is why we call it an adaptive mechanism. Therefore, it needs a mechanism that can switch between the two modes of the IDS system. For that purpose, the data packets are classified to two types. Each type work with one mode, i.e. AA packets associated with AACK mode and TA packets associated with E-TWOACK mode. To do that, we used one pit from the reserved field in the DSR fixed portion header; figure 3.4 illustrates the DSR header format as shown in the internet draft of DSR [Johnson 2004].

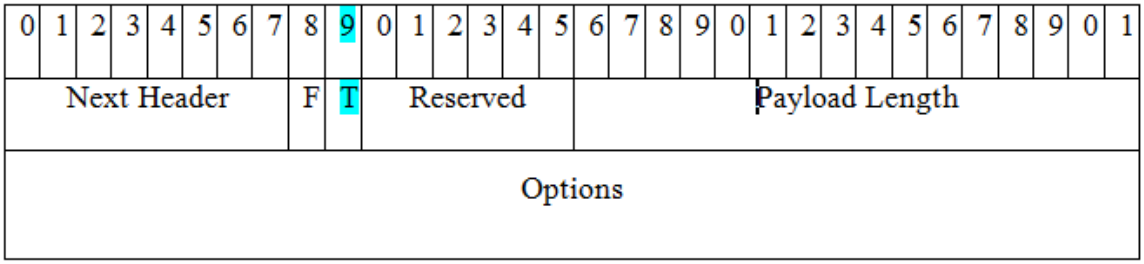


Figure 3-4: DSR header format with the packet type bit T

As illustrated in the figure 3.4, pit number 9 (T) from the reserved field is used to classify the data packets. In our implementation, we assigned a value of 1 to T to identify the TA packets (E-TWOACK mode), and assigned a value of 0 to identify the AA packets (AACK mode), thereby, for example, the forwarder node can check this bit to see if the packet is AA or TA and responds accordingly, as discussed in the previous section. The source node is the only node that can change this bit depending on the mode of the source, and it applies this to all the data packets that it sends.

The source and destination node are responsible for switching mechanism in the AACK IDS, which means that the forwarder node will not do any type of control to the switching mechanism. The role of the destination is to initiate and send a switch packet to the source node when it receives a TA packet, as described in the previous section. The switch packet is a small DSR packet which carries a unique value that will be recognized by the source node. At the other side, the source node will start any session or flow by setting this bit to 0, which means that it will start in the AACK mode. Two cases that enforce the source to change its mode:

1. Timeout of AA packet without receiving AACK acknowledgment from destination will change the mode from AACK to E-TWOACK.

2. Receiving a switch packet from destination will change the mode form E-TWOACK to AACK.

To simplify the understanding of the dynamic switching during the communication between the source and the destination, we will describe two situations. The first, when there is a malicious node in the path. The second, when the packet dropping occurs because of an error (non-intended), such as buffer overflow or collision, rather than maliciously dropped (intended). In first situation, where the malicious node exists on the path, using the scenario in figure 3.1, suppose the malicious node is N2. When source begins the session of sending packets to the destination in AACK mode, it will send the AA data packet after registering the packet id and sending time. The malicious node N2 will drop the packet, and the source will wait for a period of time (AA timeout) if no AACK acknowledgment received it will switch the mode of sending packets to E-TWOACK in which it will detect the malicious node.

In the second situation, where the packet will be dropped due to an error, also the source will act as in the first situation and change the mode to E-TWOACK, but here the first packet of TA data packets will be received by the destination because there is no malicious node in the path. When the destination node received a TA packet type, it will know that there was an error cause packet dropping and change the source mode; thereby it will send a switch packet to tell the source to go back to the AACK mode. Figure 3.5 will clarify the switching process at source node.

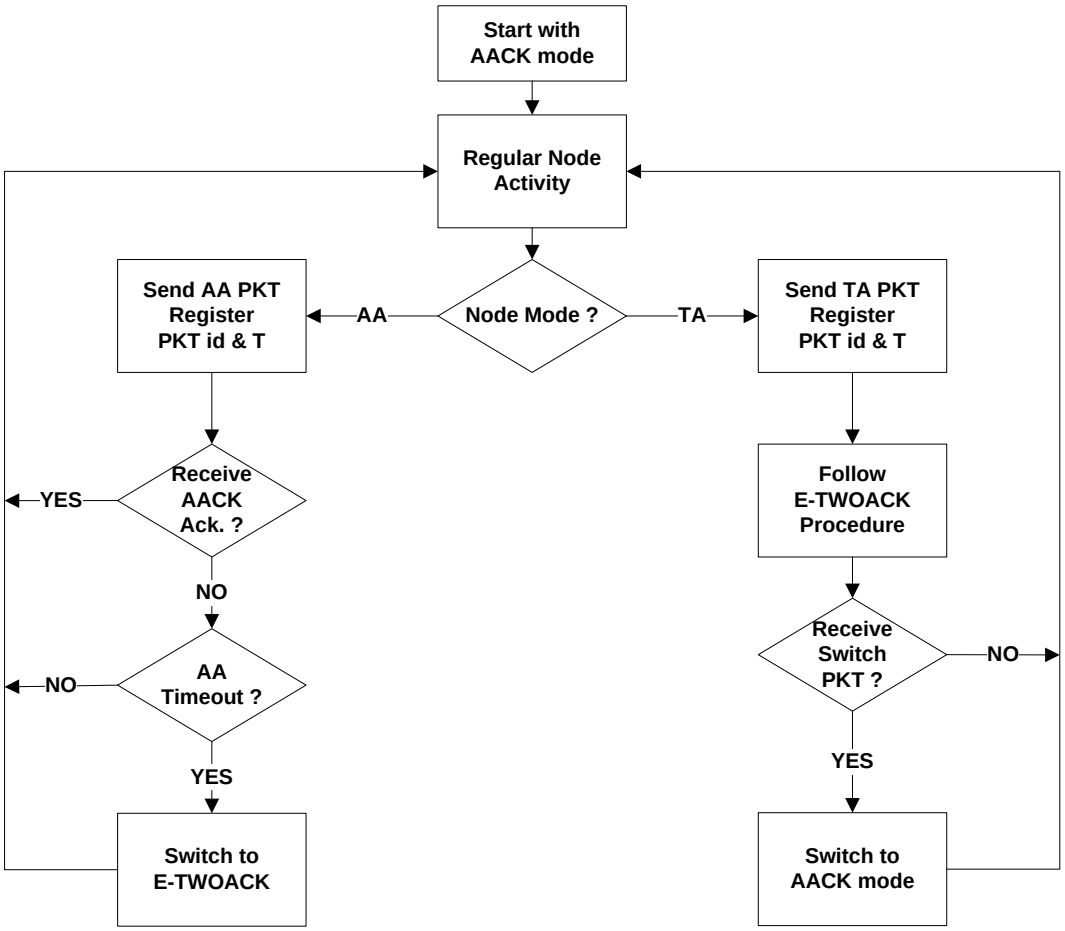


Figure 3-5: Dynamic Switching Procedure at Source Node

3.3.3 Malicious Node Model

In this study, we use the terminology malicious node to represent both selfish and malicious nodes because all of them work as a regular node and cooperate with other nodes in the network in route discovery phase, where it exchanges the routing information with its neighbors to find a route to a destination. Selfish node does that to learn more about the network to benefit itself. In addition, the main function of malicious node (selfish and malicious) is to drop all the data packets that pass through it. It will also drop all alarm packets than could pass through it in order to protect itself. Furthermore, we take in our consideration the presence of smart attackers which can exploit the

receiver collision and limited transmission power vulnerabilities in watchdog system; thereby we mix the malicious nodes in our simulation with fixed percentage of 40% with smart attackers and the rest with regular attackers.

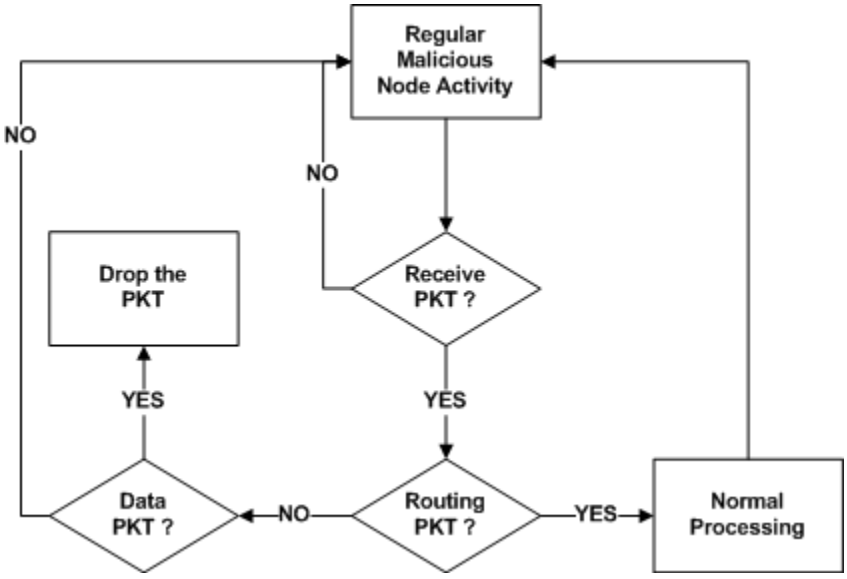


Figure 3-6: Malicious Node Procedure

3.4 Watchdog Model

In this research, a watchdog mechanism is studied for comparison with the TWOACK and AACK mechanisms.

It works as the passive acknowledgment in DSR [Johnson 2004], where each source or forwarder node, after sending a data packet, will stay listening to the next hop to verify if it will forward the received packet or not. If the next hop node does not forward the packet within the specific time (timeout), the node which is watching the next hop node will mark this node as suspicious by increasing its warning counter. If this counter reaches a specific threshold, the node marked as a malicious node and an alarm is sent to the source node. After receiving the alarm, the source node will purge its route

cache from all paths that contain the malicious node. Moreover, it will change the path to the destination and send the rest of packets over the new path. . Figure 3.7 demonstrates the pseudo code of watchdog implementation [Tanapat 2008].

Two important parameters are adjusted experimentally in our implementation. The first one is the timeout parameter and the second is the maximum threshold. There is a tradeoff when we adjust them. When we adjust a large value it will give the malicious node more chance to drop packets, whereas if we adjust them with small value it will increase the accusations of the innocent nodes (false alarms).

```
/* Watchdog Pseudo-code */

If (source or forwarder) overhear a data packet
{
        If matches recently sent packet
        {
                Record as a forwarded packet
                Status (next hop) = good
        }
        If sent packet Timeout
        {
                If count (non-forwarded pkt) > threshold
                {
                        If status(nexthop) !=good
                        {
                        Send alarm packet to source
                        Status(next hop) = malicious
                        } // not good
                } // count non-forwarded
        } // Timeout
} // overhearing
```

Figure 3-7: Watchdog Pseudo Code

3.5 TWOACK Model

We have implemented the TWOACK model to compare it with our proposed mechanism. In addition, we use it with some improvements, discussed in section 4.1, in our mechanism; thereby, we will describe it in more details in this section. For more information about TWOACK mechanism, refer to [Balakrishnan2005]. Figure 3.8, describes its procedure step by step. Each three consecutive nodes, e.g. N1—N2—N3, will apply the TWOACK acknowledgment based procedure. In the figure 3.8, in the first step the source generate a data packet and send it to the first forwarder, this node will not send any acknowledgment because it is not two hops away from the source, as show in second step the first forwarder just forward the packet to the next hop. In third step, the second forwarder receives the data packet and does the following steps:

1. Generates an acknowledgment packet, which is called a TWOACK packet, carrying the received data packet id.
2. Sends this TWOACK packet in the opposite direction of the data flow to the node located two hops back, in this case the source node, by extracting the route from the source route of the data packet.
3. Checks if the next hop is the destination node it will just forward the packet, if it is not it will register the packet id and its sending time.

This procedure, which is done by the third forwarder, will be done by the rest of forwarders along the path. Figure 3.9 shows the data structure that is used by the sender or forwarder to register the sent packets.

N2	N3	C_{MIS}	List
Next Hop	Second Hop	Misbehavior Counter	Data Packet IDs awaiting TWOACK and sending Time

Figure 3-8: Data structure of the registered information of sent data packet

In each three consecutive nodes along the path, e.g. [S … ***N1 N2 N3*** … D], N1 will register the link N2-N3 in its data base as shown in figure 3.9. It will register the next hop, N2, and the second hop, N3, and associate with it the misbehaving counter, $C_{MIS,}$ and list by all data packet IDs that are sent by N1 through the link N2-N3 and waiting for TWOACK packet. In addition, it will stamp every data packet id by the sending time, to calculate the timeout.

Detection algorithm is described briefly in figure 3.10. As mentioned above each link is associated with a misbehaving counter $C_{MIS.}$ As in the example above, N1 increases the misbehaving counter C_{MIS} every time the packet id remains in the list for a period of time more than pre-specified *timeout* without receiving a TWOACK packet from N3 that acknowledge the data packet receiving at node N3. This counter is increased until it reaches a specific limit, which is called a *misbehaving threshold.* If C_{MIS} *counter* exceeds the *misbehaving threshold*, the whole link will be considered as a misbehaving link and an alarm is sent to the source node.

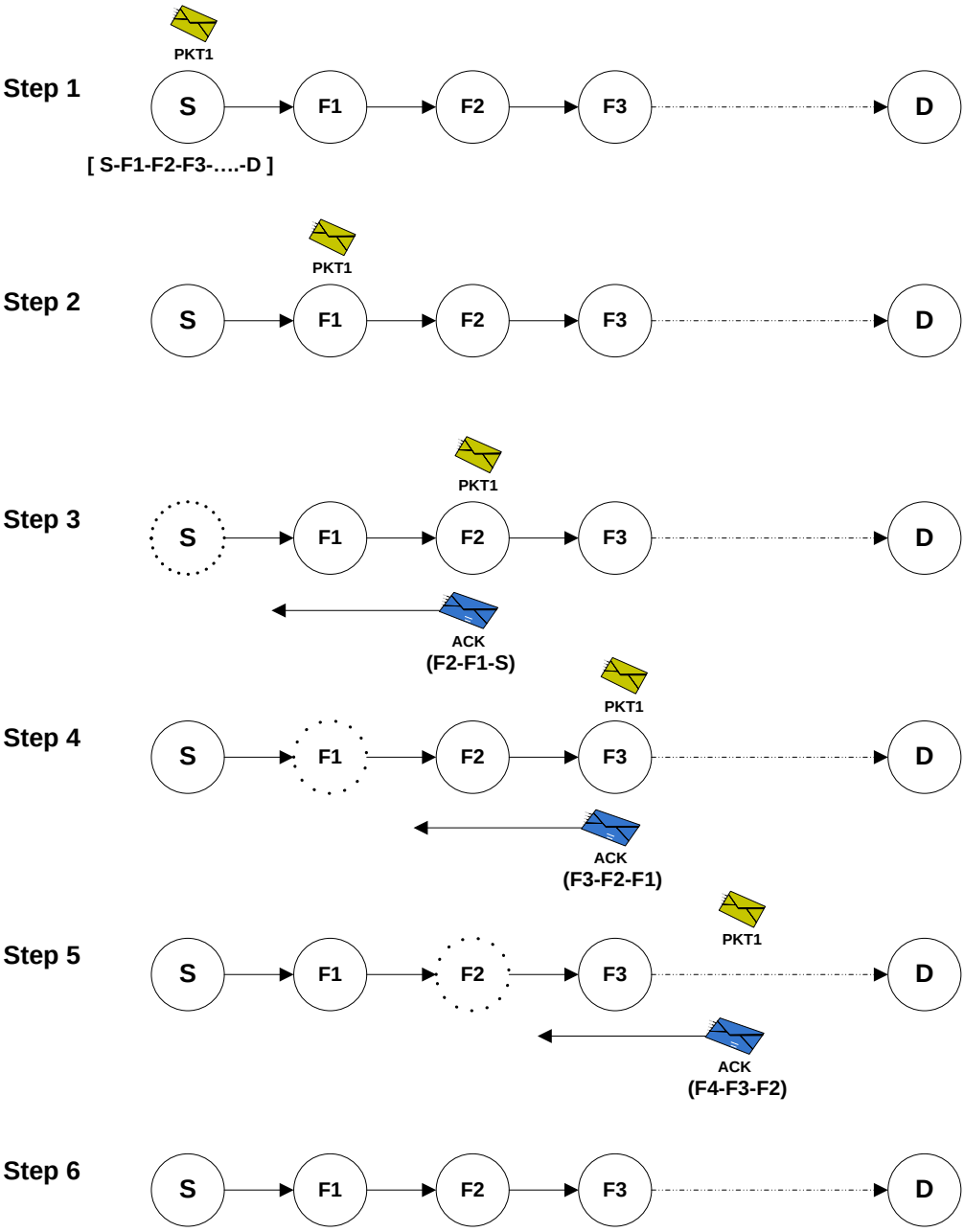


Figure 3-9: TWOACK Mechanism Description, Tracing one packet travels along the route from source to destination

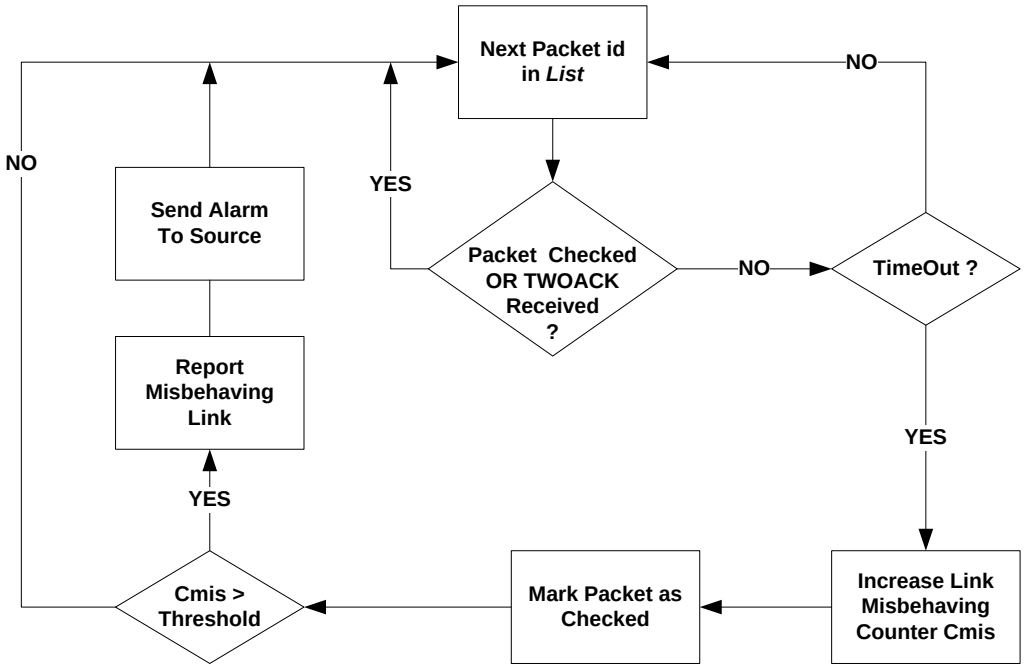


Figure 3-10: TWOACK Detection Procedure

3.6 AACK Model

In this section, we will describe our proposed mechanism in more details. As we have discussed in the beginning of this chapter, ***AACK*** is an abbreviation of ***A***daptive ***Ack***nowledgment, which composed of two main schemes. First scheme, will be described in section 3.6.1, is the ***E***nhanced ***TWOACK (E-TWOACK)***; and the other is the AACK, will be described in section 3.6.2, which is the end to end acknowledgment scheme. Note that it has the same name as the composed mechanism ***AACK***.

3.6.1 E_TWOACK Model

E_TWOACK is an abbreviation of ***E***nhanced ***TWOACK***, where it works as the famous TWOACK mechanism except that we improve it by adding a mechanism to

detect the malicious nodes rather than links. As we have discussed in the previous chapter, one of the disadvantages of the TWOACK scheme is that it detects a misbehaving links instead of detecting the misbehaving nodes, and that will give the malicious node more chance to drop more data packets, for example, suppose N1 is a malicious node and it has many links in the network, e.g. it is exist in path1 (S->***N0->N1***->N2->D) and in path2 (S->***N4->N1***->N3->D). In this example, the TWOACK mechanism reports N0->N1 and N4-N1 as misbehaving links, and each one of them will be considered as a different entry to the misbehaving links data base; thereby, each link has a chance to drop packets until it reaches the specified threshold, that will double the total dropped packets by malicious node N1. Furthermore, there would be many links in the network that contains N1 as one of its ends, and that will increase the dropped packets much more.

For that reason, in our mechanism, we improve the procedure of the TWOACK to make it more accurate in the detection of malicious nodes. All intrusion detection systems that have been proposed assume that the malicious node exists at the intermediate nodes; thereby, we all agree that it will not be a destination node. In our proposed solution, we use this assumption to determine the misbehaving node in the links that have a destination node in the other end, e.g. N1->D.

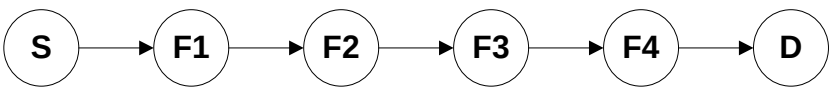


Figure 3-11: Example of the three types of nodes in the routing path (Source, Forwarder, and Destination)

Figure 3.8 explores a general example of a path from source to destination in the source route. There are four possibilities of the three consecutive nodes along the path. These possibilities are as follows:

- Case 1: S-F1-D (source then forwarder then destination)
- Case 2: F1-F2-D (forwarder then forwarder then destination)
- Case 3: S-F1-F2 (source then forwarder then forwarder)
- Case 4: F1-F2-F3 (forwarder then forwarder then destination)

In the regular TWOACK mechanism, the first node in the four cases, which is S or F1, will report that F1-D, F2-D, F2-F3, and F1-F2 links respectively are malicious links. Unlike the TWOACK mechanism, E-TWOACK will detect the exact malicious nodes as follows:

- In the first two cases, the first node, which is S in (S-F1-D) or F in (F1-F2-D), will report that the node just before the destination is a malicious node. For example in case one F1 and in case 2 F2 will be reported as a malicious node.
- In case three, S-F1-F2, suppose F1 is the malicious node then S will know that because there will be no acknowledgment or alarm received by S. on the other hand, if F2 is the malicious node F1 will send alarm to S then S will detect that F2 is the malicious node.
- In case four, node F3 will be reported as a malicious node by F1. That is because the node before F1, which is could be F0 or S, has reported that the link F1->F2 is a good link because it has received TWOACK packets from F2; thereby, F3 is the one who drops the packets.

The E-TWOACK has the same data structure, shown in figure 3.9, as TWOACK except that there will be no second hop. Likewise TWOACK, each misbehaving node will have the misbehaving counter, $C_{MIS,}$ that will be increased each time timeout occurs, and if this C_{MIS} exceeds the specified threshold, the misbehaving node will be considered as a malicious node and an alarm will be send to the source node.

3.6.2 Aack Model

In this section we describe the second part of the complete IDS AACK, which has same name as the complete system. *Aack* is an end to end acknowledgment based scheme. We distinguish it from the complete system by writing the ACK in small letters, e.i., Aack.

In this scheme all the intermediate nodes work regularly without any change in their functionality. This appears an important advantage of our mechanism where the intermediate nodes will not consume their power and memory all the time, and these two resources (power and memory) are very important and scarce resources in MANETs, as we have discussed in the first chapter. In this scheme, the source and destination work with each other to ensure delivery of data packets. The destination will acknowledge each received data packet with *Aack* acknowledgment that will extract its route from source to destination from the source route of received data packet.

That means we have two types of acknowledgment packets in our mechanism:

- *Aack*: This is from source to destination acknowledgment, in AACK mode.
- *TWOACK*: This is between each three consecutive nodes, in the TACK mode.

Thus, the Aack acknowledgment will travel in the opposite direction of the data flow to the source node. This process continues until a timeout happens at source. The timeout (at source node) is calculated as in equation 3.1, it is not a fixed value. It depends on the number of hops in the path and the *Tack* mode timeout.

$$\text{Aack}_{\text{Tout}} = \frac{\text{Tack}_{\text{Tout}}}{2} \times \text{No. of Hops} \quad (3.1)$$

Tack timeout is determined experimentally, and more details about it will be described in the next chapter. Because the Tack timeout takes in account a number of two hops, we divide it by 2 to get the timeout of one hop transmission, which is a constant value whereas the no. of hops is a variable value.

When timeout occurs, the source uses the switching system, as described before in this chapter, to switch from Aack mode to Tack mode. Where in this mode will work as the TWOACK mechanism until it detects a misbehaving node or discovers that it was an error, by the arrival of a data packet of type TA, Tack mode, to the destination. If TA data packet arrives to the destination, that means there is no misbehaving node in the path from source to a destination. Therefore, the destination will send a special packet named a switch, which tells the source to come back again to an Aack mode, and so on. If there is a malicious node, it will be detected as described in the *E-TWOACK* section.

3.7 Response System Model

When the misbehaving node is discovered by the detection system, the response system will be triggered by the same node that detects the misbehaving node. All nodes in the network are capable of doing detection and response functions. And all nodes have an array to register the good and bad nodes based on the behavior of that node in the network.

When the node discovers a malicious node it will inform the source node by sending an alarm, which is a small packet that is generated by the routing protocol and extract the route from the source route of the data packet. This alarm will carry the malicious node id. Each node in the path will forward the alarm and learn from it about the malicious nodes. Furthermore, they will add the source node of the alarm as a good node in their data base. In addition, all nodes that will overhear the alarm will learn from it about the malicious and good nodes.

4 METHODOLOGY AND PERFORMANCE EVALUATION

In this chapter we describe our methodology that we follow to evaluate AACK mechanism. We evaluate our proposed mechanism by means of simulation. A brief description for network simulator NS2 that we have used in our simulation is given in this chapter. This chapter includes two main sections; the first is the simulation environment, and the second is the results analysis.

4.1 Simulation Environment

One of the main challenges of this research was the NS2 simulator with its complex interface. NS2 version 2.33 was used with fedora 8 Linux platform and desktop computer with Pentium 4 CPU with speed of 3.8 MHZ and 1 Giga RAM. This version does not supported with video traffic; that takes us more time to search for contributions of NS users to add video traffic to the NS simulator. Two main contributions that we have used; one of Dr. Ashraf Mattrawy [Mattrawy 2002], and the other was for Chih Heng [Chih 2007]. Also, we used some scripts from Evalvid-RA [Lie 2007].

4.1.1 Simulator Description

We use Network Simulator-2 (NS-2) version 2.33 in our simulations because it is more flexible to change its internal files, open source, and free. For that reasons, it is considered a good simulation tool for researchers; it is the most popular simulator used by the mobile ad hoc network researchers [Kurkowski 2005], see figure 4.1 that shows a survey of simulation-based papers in ACM's international Symposium on Mobile Ad Hoc networking and computing (MobiHoc) 2000-2004. Until now it depends on the contributions of the researchers. A good brief description for NS2 will be found in

[Palaniappan 2005] and for more details it will be found in the NS2 full documentation in [Fall 2003]. The simulator is written using two object-oriented languages, C++ and OTcl. The C++ compiled components run the core simulation engine, event schedulers and agents. The OTcl based interpreter is used to setup the simulation configuration and controls of the C++ data path.

The dual design benefits from the execution speed of the C++ compiled network objects and rapid reconfigure-ability of interpreted OTcl configuration objects. Most often in simulation studies the parameters change with every new simulation, but the underlying protocols and data agents remain the same. Therefore, it is useful to have a rapidly reconfigurable simulator as the basis for using the dual interpreter/compiled class hierarchy. Since OTcl are interpreted changes in simulation parameters do not have to be recompiled, a researcher can run large sets of simulation with a one-time compilation of the C++ network objects. The control parameters and functions of the C++ compiled objects are exposed to the OTcl interpreter via OTcl linkage. For every OTcl object invoked in the interpreter hierarchy there is a mirrored object created in the C++ hierarchy. NS2 can be downloaded from [NS].

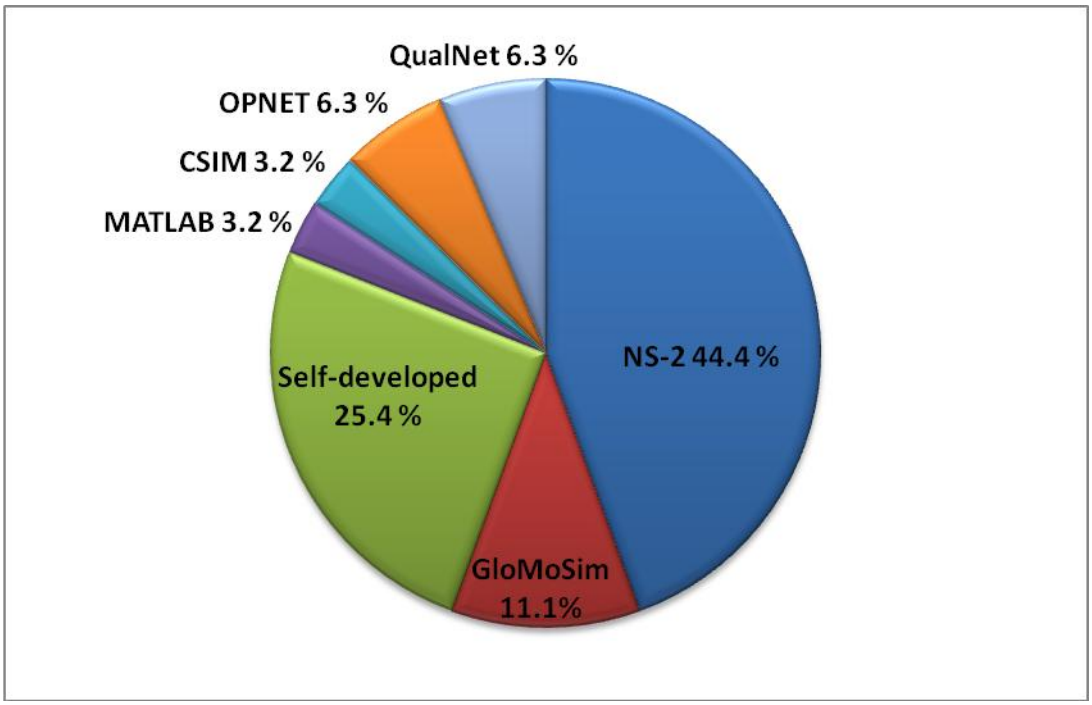


Figure 4-1: Simulator usage survey of simulation-based papers in ACM's International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2000-2004 [Kurkowski 2005].

4.1.2 Simulation Scenarios

AACK mechanism is examined under different conditions. We change the mobility speed and traffic parameters to get different scenarios. High and low mobility simulations are conducted with two types of traffic, which are the constant bit rate and video traffic. The two mobility speeds are 1 meter/second and 20 meter/ second, these two speeds represent the pedestrian and car motions. 50 mobile nodes are deployed over space area of 670 X 670 meter square. These mobile nodes classified to three categories:

- Normal nodes: these nodes do regular operations including the IDS in the case of the IDS mechanism.

- Attacker nodes: these nodes apply the regular attack model that just dropping data packets.
- Smart attacker nodes: they do the same functions of the attacker nodes with the ability to cheat the monitor by using power control and intended collisions with its next hop, as described in the previous chapter.

In all scenarios the three categories are exist in different percentages. The number of attackers varies from 0 % to 40 % from the total number of nodes, which are 50 nodes. Smart attackers form 40% from the total number of attackers. The number of runs was 10 run per one data point and the average is taken. Each run has different mobility scenario, that means different initial positions and different destination locations, and different seed number, the seed number in NS2 is a variable that change the random variables that is generated and used in each run. Figure 4.1 shows an example of one random initial mobility scenario.

The traffic that we have used in our simulation can be classified into two types: low traffic (CBR) and high traffic (Video). In video traffic we build a small scenario of 5 nodes to examine performance of video transmission of both types MPEG4 and H.264 over DSR routing protocol to see the performance of the protocol during both types of video traffic. Then based on the results we use the MPEG4 to evaluate our mechanism. The video traffic is evaluated over high mobility and 30 mobile nodes.

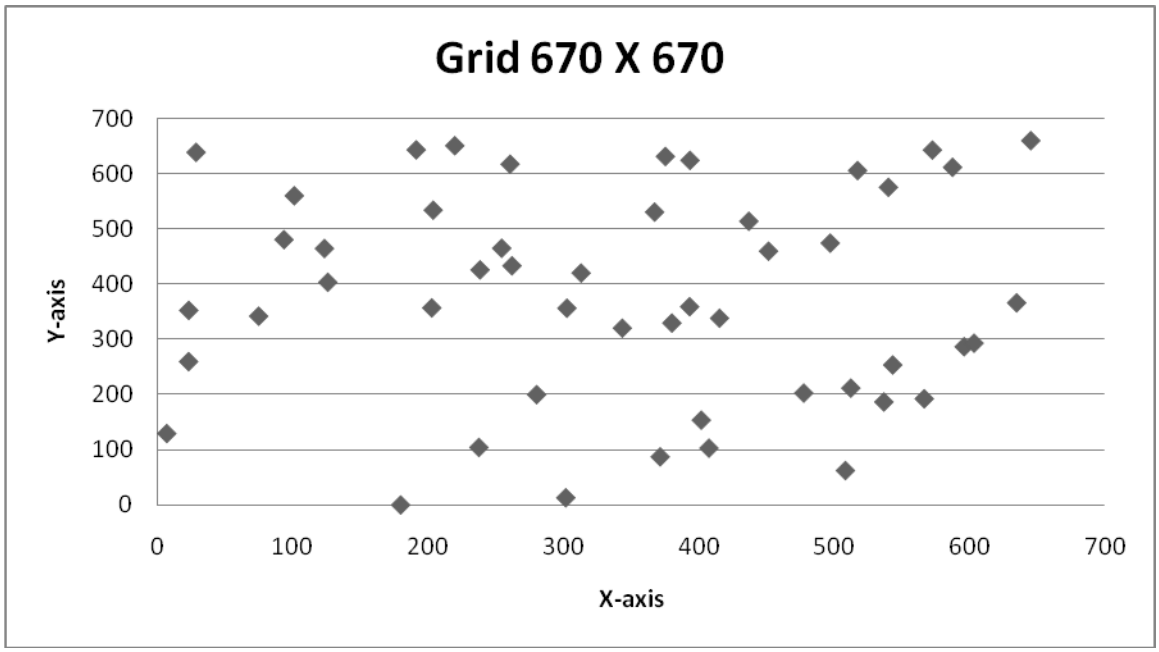


Figure 4-2: Initial Simulation Grid

4.1.3 Simulation Parameters

Table 4.1 shows the simulation parameters of DSR routing protocol that is used in our simulations. Other simulation parameters setting are shown in table 4.2.

Table 4-1: DSR simulation parameters

Parameter	Value
DSR snoop forwarder errors	ON
DSR snoop source routs	ON
DSR salvage with cache	ON
DSR use Tap (promiscuous mode)	ON
DSR reply from cache to RREQ	ON
Max. time between RREQs	10 sec
How long a packet can live in send buffer	30 sec

Table 4-2: Other Simulation Parameters

Parameter	Value	
Number of nodes	50 nodes	
Simulation area	670 meter X 670 meter	
Simulation time	900 second	
Mobility model	Random waypoint with pause time 0	
Speed rang	Uniformly distributed (0.1-20) meter/second	
Maximum speed	1 (low mobility) and 20 (high mobility)	
Traffic type	CBR	MPEG4
Packet size	512 bytes	Variable (Max, 1028B)
Packet rate	4/sec	30 frame / second
Maximum connections	10	8
Propagation model	Two-ray ground model	
Antenna model	Omni-directional	
Transmission range	250 meter	
MAC protocol	802.11 CSMA/CA	
Link Bandwidth	2 Mbps	
Routing protocol	Dynamic Source Routing (DSR)	
Watchdog timeout	0.1 sec	
Watchdog, TOWACK threshold	40 packets	
AACK threshold	30 Packets	

The timeout is a very important parameter and its value affects the efficiency of the detection system. That is because if it is very large it will give the misbehaving node more chance to drop packets before detection. On the other hand, if it is very small it will increase the false alarms, which degrade the performance significantly. In our study, we have three mechanisms to evaluate: the watchdog, TWOACK, and AACK. Watchdog overhear the next hop node to acknowledge that it has forwarded the packet, which means that its timeout is just considered for one hop away (one RTT, Round Trip Time the time from sending packet to receiving acknowledgment). Whereas in case of TWOACK it needs to acknowledge the receiving of data packet at node that is two hops away, which means it is time out bigger than in watchdog. Unlike watchdog and TWOACK, they have a constant timeout value, AACK mechanism has a variable timeout value because it has a random value of number of hops. That means, it depends on the number of hops in the path from source to destination. Parameters could be changing during simulations depend on the scenarios used.

Time out is calculated as follows:

Watchdog needs just round trip time (RTT) of one hop, which is specified experimentally by taken the average value of the time from the data packet sending time to the time when watchdog overhears the forwarding of the same packet. As presented in table 4.2, it was set to be 0.10 second. In TWOACK scheme, the packet needs to be acknowledged from the node that is two hops away from the sender. Similar to watchdog, it is determined experimentally. The period of time from sending a data packet to receiving the acknowledgment of the same data packet is determined experimentally and the average value is taken. It was set to 0.2 second.

The AACK timeout is variable; it depends on number of hops and the value of TWACK timeout. It is calculated dynamically during the simulation run using the equation 4.1.

$$\mathrm{AACK_{Tout}} = \frac{\mathrm{TACK_{Tout}}}{2} \times \mathrm{No.\ of\ Hops} \qquad (4.1)$$

4.1.4 Performance Metrics

Three performance metrics are used to evaluate AACK, Watchdog, and TWOACK schemes. They can be defined as follows:

- Packet delivery ratio (PDR): it is the ratio of the total number of received packets at the destination to the total number of sent packets by the source.

$$PDR = \frac{\sum Received\ packets\ at\ destinations}{\sum Sent\ packets\ by\ sources} \qquad (4.2)$$

- Routing Overhead (RoH): this is the ratio of routing-related packets in bytes (RREQ, RREP, RERR, AACK, TWACK, alarms, and Switch) to the total routing and data transmissions (sent or forwarded packets) in a simulation in bytes. That means the acknowledgments, alarms and switching over head is included.

$$RoH = \frac{\sum Routing\ Messages}{\sum Data\ \ transmissions + \sum Routing\ transmissions} \qquad (4.3)$$

- Average end-to-end delay (AED): the end-to-end average delay for all successfully received packets at the destination. It is calculated for each data packet by subtracts the sending time of it from the received time at final destination. Then the average represents the AED.

$$AED = \frac{\sum_1^N (T_{Received} - T_{Sent})}{N} \qquad (4.4)$$

N.B: N is the number of successfully received packets.

In video traffic evaluation we have added one performance metric, which is the end to end delay versus number of hops. That is calculated by taken the average value of end to end delay for each grope of packets in which the all sent packets will be classified into several gropes based on number of hops it takes to reach the destination.

PSNR (Peak Signal to Noise Ratio) is a performance metric that is used to compare the two codec compression techniques, MPEG4 and H.264. It measures the error between a reconstructed image and the original one. ITU-R introduced quality metric called mean opinion score (MOS), shown in table 4.3. The MOS values varying from 1 (bad) to 5 (excellent) and it depends on the peak-signal-to-noise ratio that is calculated using Eq. (4.5) [ITU 1991, Symes 2001].

$$PSNR = 20log_{10}\left[\frac{V_{peak}}{\sqrt{(\frac{1}{N})\sum_i\sum_j\left(Y_{ref(i,j)}-Y_{prc(i,j)}\right)^2}}\right] \qquad (4.5)$$

Where $Y_{ref(i,j)}$ and $Y_{Prc(i,j)}$ are the pixel value of the reference and reconstructed frames, respectively. The total number of pixels in a frame are represented by N, where $V_{peak} = 2^k-1$ and k = number of bits per pixel (luminance component). PSNR calculates the error between a reconstructed image and the original one. Mapping between PSNR values and MOS is used to recognize the quality of video stream as a human visual system [Chee 2007]. Thus, based on the PSNR value of the frame it will be classified into bad, poor, fair, good, or excellent depends on the table 4.3.

Table 4-3: PSNR to MOS mapping

PSNR [dB]	MOS value	Class
≥ 37	5	Excellent
31-37	4	Good
25-31	3	Fair
20-25	2	Poor
<20	1	Bad

4.2 Simulation Results

The simulation results are two main parts: the CBR and video traffic. In the CBR traffic there are two mobility scenarios. In the video traffic there are the comparison results of MPEG4 and H.264 video types; and the MPEG4 results that is used to evaluate AACK versus original DSR and TWOACK.

4.2.1 CBR Results

In each part, low and high speed, we separate the results into two categorize. The first one shows only the results of AACK and TWOACK mechanisms, to make the comparison of the results easier. Then the second part will include all four schemes DSR, Watchdog, TWOACK, and AACK, to compare the acknowledgment mechanisms with the DSR, and watchdog.

4.2.1.1 Low Speed

The network is more stable and dropping packets due to network errors is very low. Thus, both mechanisms AACK and TWOACK expected to be efficient intrusion detection systems. They outperform both DSR and Watchdog by around 25% in the packet delivery ratio, as shown on figure 4.4 (packet delivery ratio). Figure 4.3 clarify the

different performance of the AACK and TWOACK in terms of packet delivery ration, routing overhead, and end to end delay. As it is illustrated, in case of small number of malicious nodes, in PDR, the AACK slightly outperform the TWOACK, whereas in case of large number of misbehaving nodes (30% and 40 %) the performance of AACK is better than TWOACK by approximately 6%. In average the routing overhead of AACK is less than TWOACK due to the reduction of the TWOACK packets that will not work all the time as discussed before. The end-to-end delay of AACK is less because it reduces also the computation time of the intermediate nodes. the routing overhead of both AACK and TWOACK increases with the increasing of misbehaving nodes that is because of the high detection rate that will increase the routing discovery phases that increases the RREQ broadcasts which increase the routing overhead.

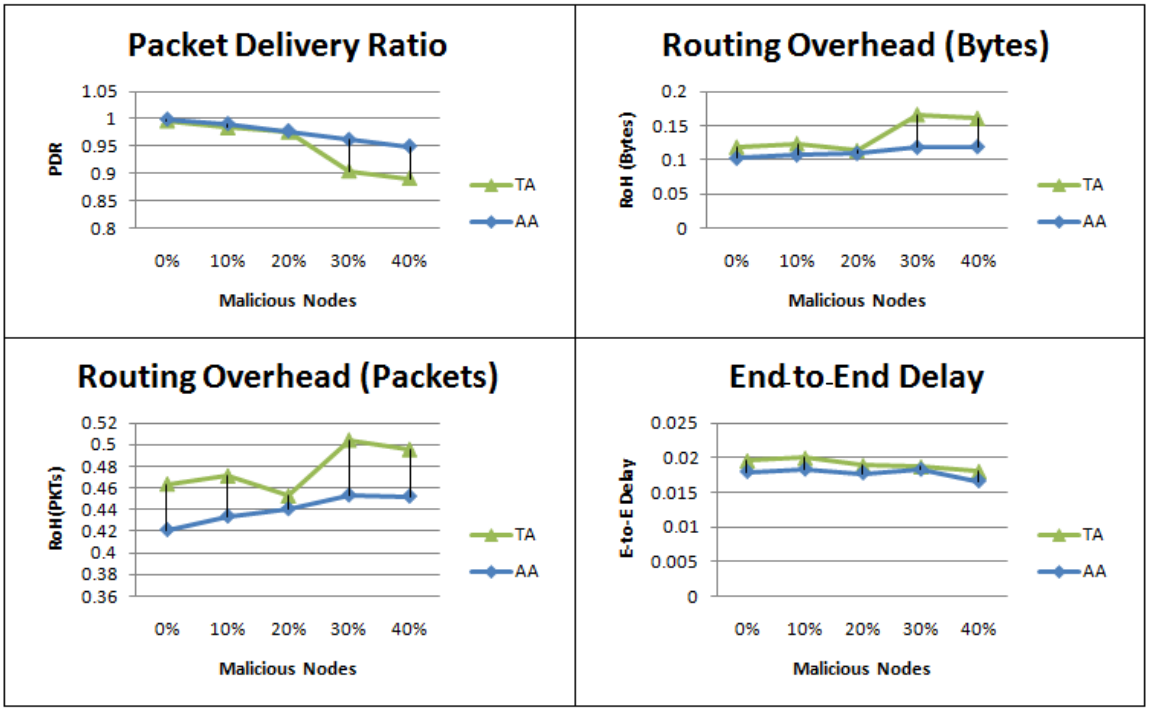


Figure 4-3: Low speed results of AACK and TWOACK comparison

In figure 4.4, we add the DSR and Watchdog to compare their performance to the acknowledgment based mechanisms, AACK and TWOACK. PDR of the acknowledgment based mechanisms outperform the DSR and Watchdog approximately by 25 % while this cost it about 10 % overhead. Because of the partial detection of watchdog, just detects the regular attackers, it outperform the DSR by 5 %. The overhead of DSR and watchdog is almost the same because the detection mechanism of the watchdog depends on overhearing, which just consume more power and computation time; thereby, there is no additional control packets used for detection system, e.g. acknowledgments. Also, notes that the end to end delay of the DSR and Watchdog decreases while the misbehaving nodes increases that is because the misbehaving nodes cover the network errors such as broken links this will fool the source that the path is working well so the shortest paths still works and there will be no longer paths that increase the end to end delay. Furthermore, the packets that are dropped are not included in the calculation of the end to end delay.

The end to end delay of the TWOACK is the largest. That is because it has the TWOACK packets that increases the collisions due to it is opposite direction to the data flow and also due to the computation time of the detection system in all the nodes along the path.

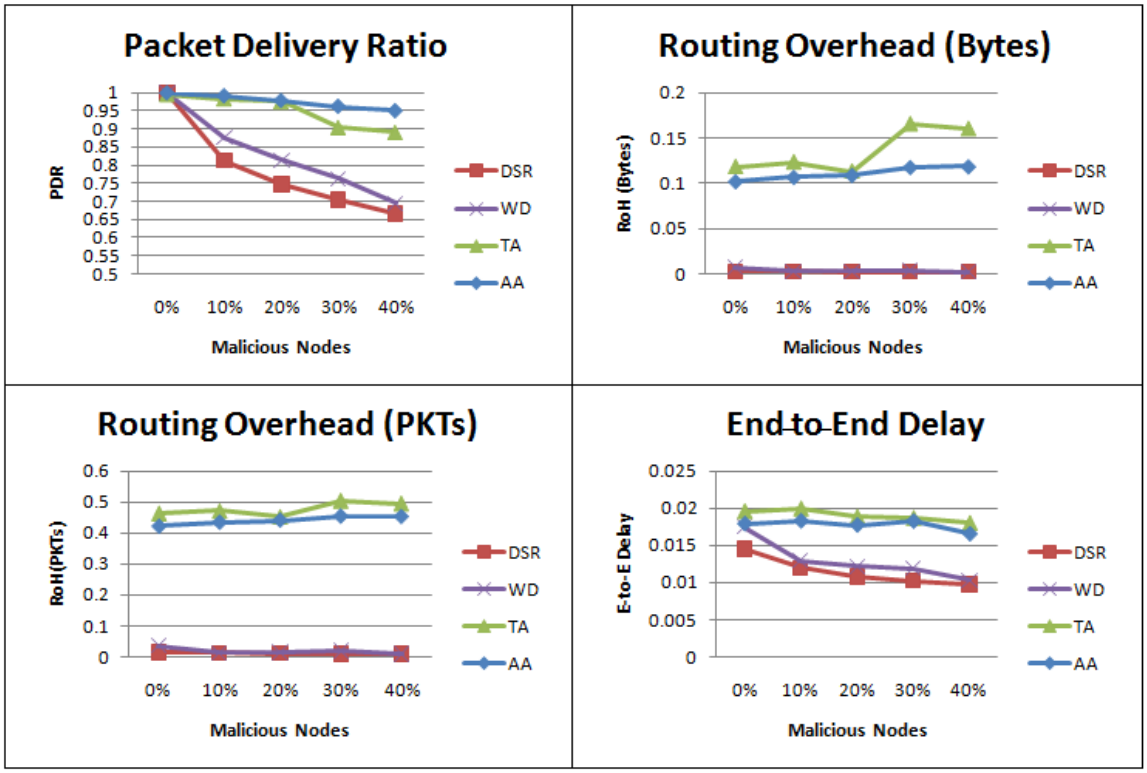


Figure 4-4: Low speed results of the four schemes DSR, WD, TWOACK, and AACK

4.2.1.2 High Speed

Due to the effective detection of misbehaving nodes in AACK rather than misbehaving links in TWOACK, and also due to the less overhead acknowledgment packets the AACK outperforms the TWOACK in high speed scenario by approximately 15 % in case of the 0% to 30% of misbehaving nodes in the network; but in case of 40 % the TWOACK has better performance that AACK that is could be because the switching overhead of AACK. Because the broken link rate is high in the high speed scenarios, thus the AACK will keep switching between AACK and TWOACK every time it encounters a broken like error that will increase the switching overhead packets, and that also explain

the increasing of the overhead of AACK in the routing overhead result (in case of 40% misbehaving nodes).

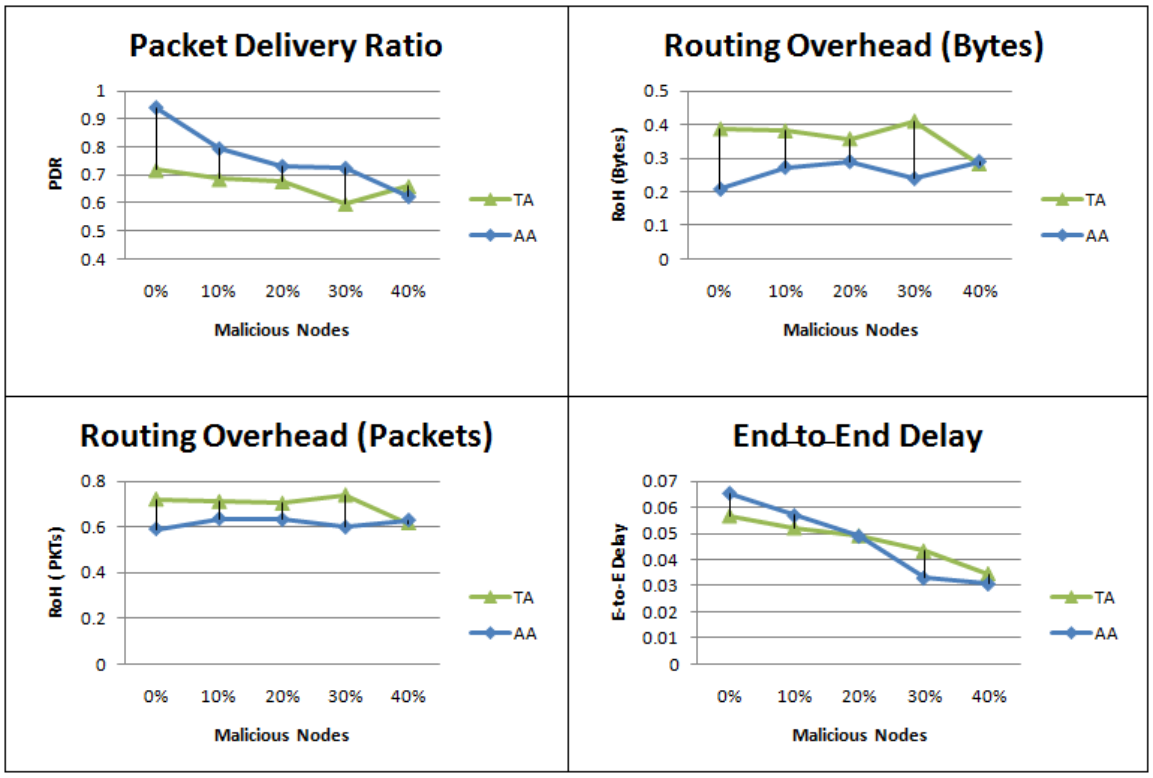


Figure 4-5: High speed results of AACK and TWOACK comparison

In figure 4.6, the PDR of the four schemes at 0% misbehaving nodes it is more clear hear than in the case of low mobility that the DSR and Watchdog outperforms the acknowledgment-based schemes due to that they have more overhead especially with the overhead of high mobility which increases the broken links rate. the overhead of the acknowledgment based schemes here is more than in case of low mobility by about 10 %; thereby, as the figure illustrates it is larger than the overhead of watchdog and DSR by approximately 25 % while the increasing in the performance about 10 %. That means the benefits of the acknowledgment-based schemes is more beneficial in cases of low mobility scenarios. The end-to-end delay of the TWOACK and AACK on average is

almost the same because in high mobility as we have mentioned before the overhead of the switching scheme add more computation time.

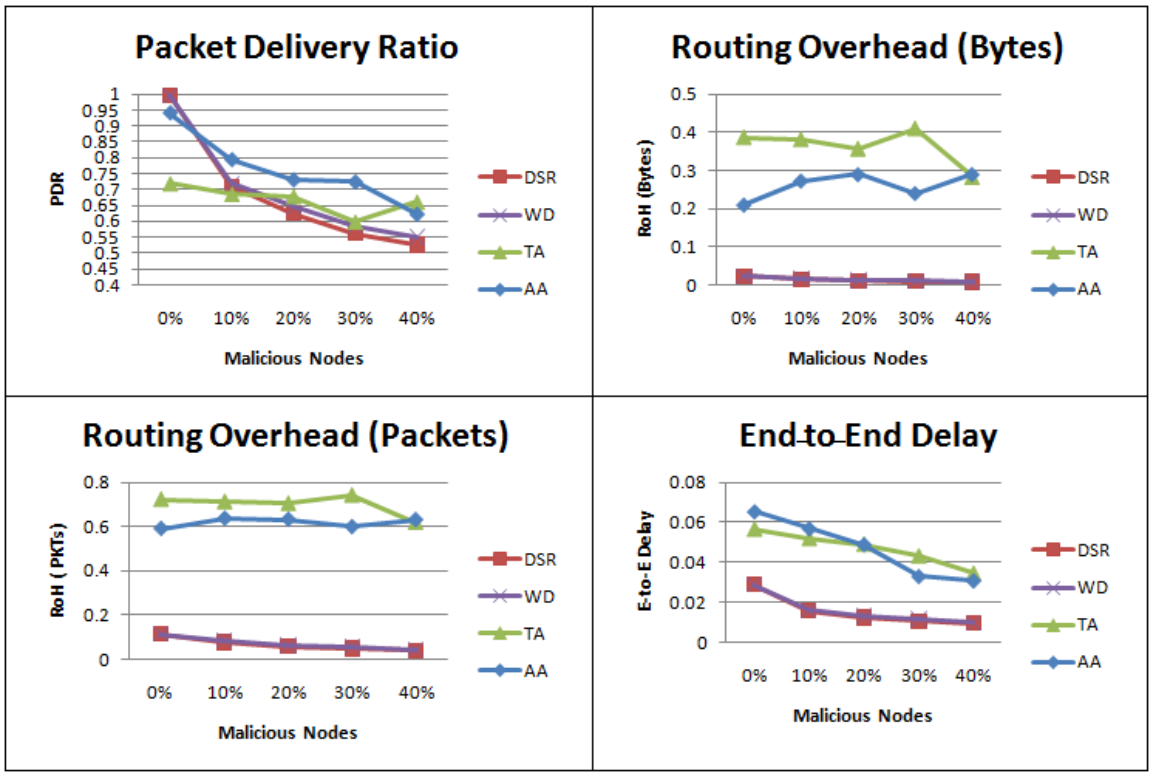


Figure 4-6: High speed results of the four schemes DSR, WD, TWOACK, and AACK

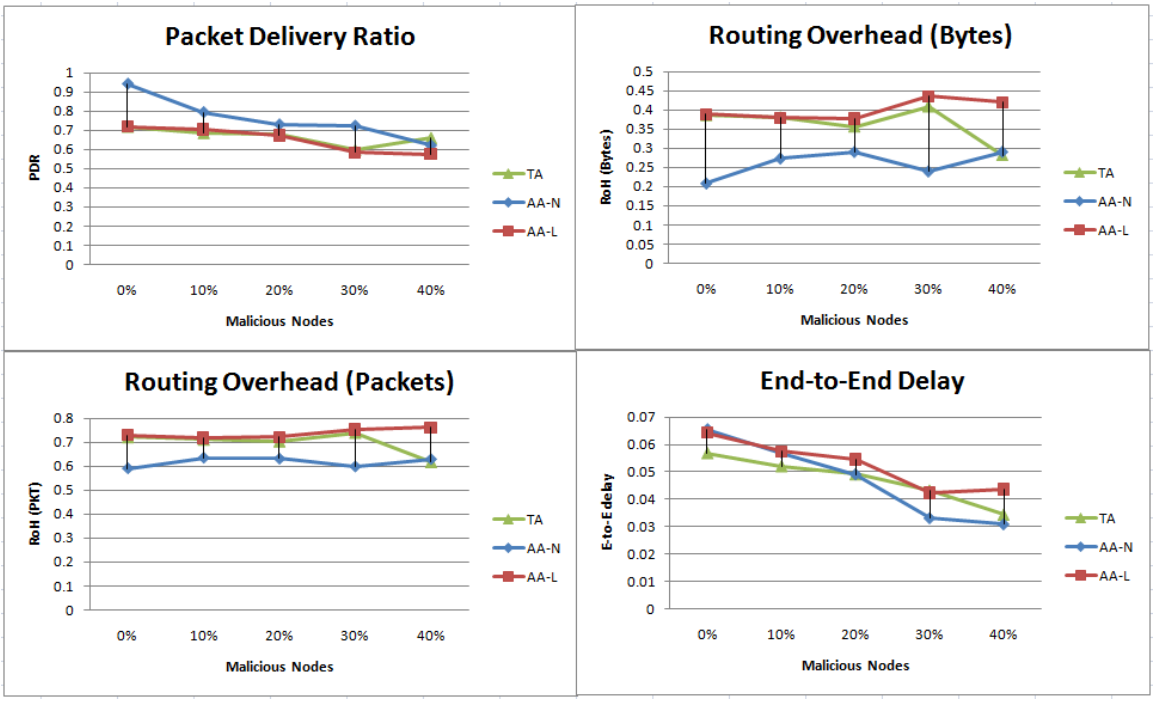


Figure 4-7: Node detection enhancement

Figure 4-7 shows the effect of the detection efficiency of TWOACK by applying node detection instead of link detection. AA-N curve indicates the Adaptive ACK with node detection and AA-L indicates Adaptive ACK with link detection. The AACK with node detection outperforms both TWOACK and AACK with link detection because it detects the exact misbehaving node. Whereas the TWOACK and AACK with link detection are closed to each other because in high mobility the switching overhead is high and the TACK mode is working most of the time.

4.2.2 Video Results

We use the video traffic to apply our new mechanism on real applications. To our best of knowledge, and throughout the literature review, this is the first time to apply video traffic over intrusion detection systems over MANETs.

4.2.2.1 MPEG4 and H.264 Comparison

A small experiment is conducted to examine the MPEG4 and H.264 using to see what is more appropriate for examining DSR routing protocol in NS-2, for purpose of using it in our simulations. This simulation includes 5 stationary nodes distributed around space area of 500 meter X 500 meter. Frame rate was 30 frames per second and each frame will be fragmented into 1000 bytes for transmission. (Maximum packet length will be 1028 bytes, including IP header (20bytes) and UDP header (8bytes)).

The foreman pictures in figure 4.3 include three types of frame groups: the MPEG4 output frames that is decoded and received at destination, the original frames that is sent by the source, and the H.264 frames that is also decoded and received at destination. From those pictures we can note that the quality of MPEG4 is better than H.264. Furthermore, the result shown on figure 4.4, the PSNR values of MPEG4 and H.264, could be explained to be more understood by using table 4.3 to map the results into the equivalent MOS value. We conclude that on average the MOS value for MPEG4 is good whereas in case of H.264 it is fair. Thus, the MPEG 4 is more suitable for our simulations than H.264, which is because it gives better results in terms of quality and PSNR.

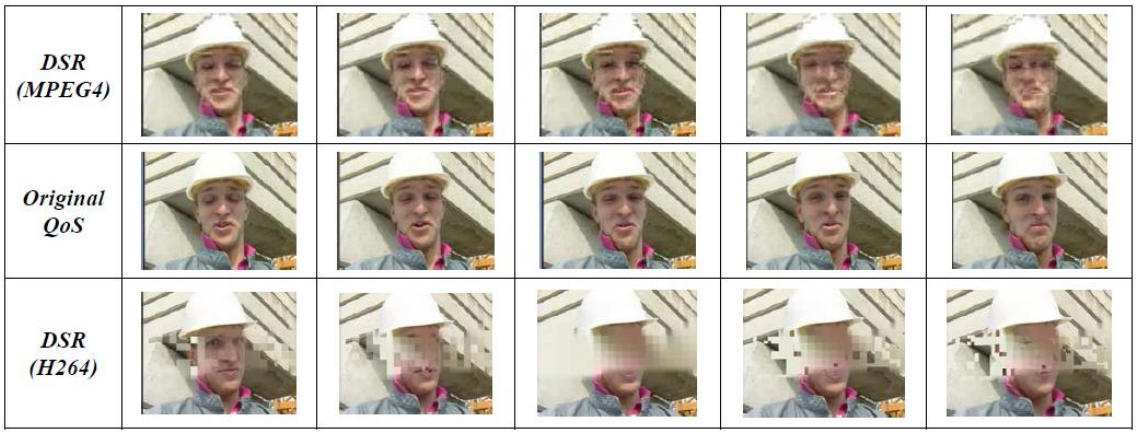


Figure 4-8: The frames quality comparison of MPEG and H.264

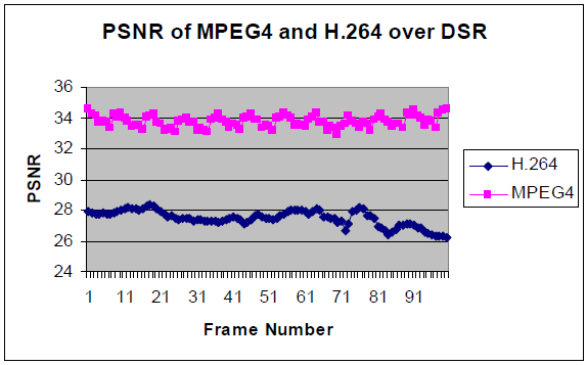


Figure 4-9: PSNR graph

We use the MPEG4 video generator that is based on TES model [Mattrawy2002], which was the more flexible for our scenarios. In video scenarios, we use 30 nodes with high mobility (pause time = 0, and speed = 20 meter/sec), the grid is the same as in the CBR traffic, 670 x 670 m^2. The video traffic experiment lasts 200 sec. The simulation time is less than in CBR scenarios, because the video traffic generators in NS-2 simulator are still not supported by developers. All what is used until this date are just contributions from others and these contributions still do some exception and segmentation errors. That is why we reduce the simulation time of video traffic to 200 sec.

4.2.2.2 MPEG4 Results

In figure 4.10, DSR has a PDR value around 35 percentage when there are no misbehaving nodes, that is because we used the worst scenario in the video traffic, where it has high mobiltiy, with pause time of 0 and uniformally distributed speed 1-20 meter/sec, and high traffic load with 30 frame/sec. whereas in [Pandian 2006], the corresponding value for DSR was around 50 % because it has less number of nodes, 20 nodes, and less traffic load which is 4-10 packet/sec. the AACK outperform the DSR and TWOACK in persence of misbehabing nodes by approximately 2% the performance enhancement here is the lowest because it is high mobility with high traffic load. Note the routing overhead in figure 4.11. In the high speed of CBR the TWOACK reaches around 40% routing overhead and AACK reaches around 30 %, whereas here in case of video traffic the overhead is much less than in case of CBR.

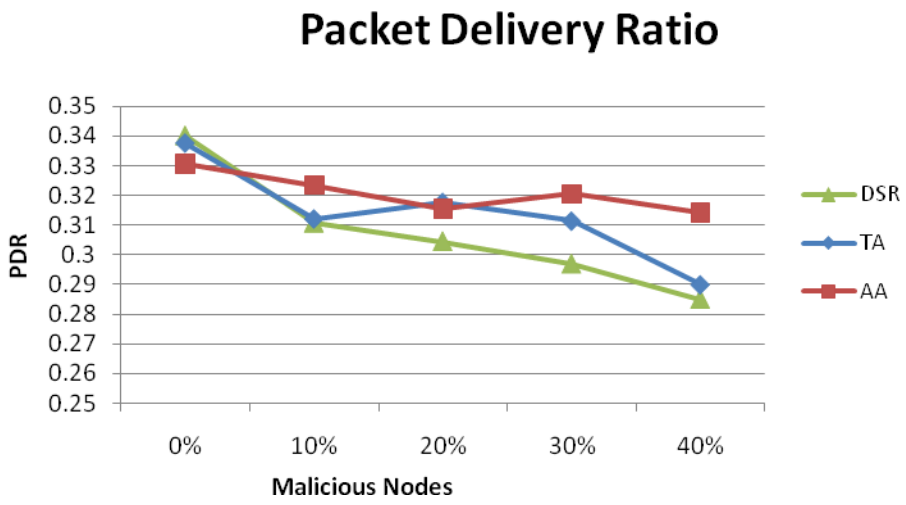


Figure 4-10: Packet Delivery Ratio of MPEG4 over DSR with Misbehaving Nodes

The routing overhead in this case for the TWOACK on average 4 % and for AACK it is almost the same, this is because the traffic here is larger than in CBR where it is 4 packet per second in case of CBR and here in case of video traffic it is 30 frame per second and each frame could be fragmented to number of packets. According to the equation of 4.3 the larger data traffic results in smaller routing overhead. So the effect of increasing the control packets, routing packets, will not be clear in case of huge data traffic especially here with small simulation time period.

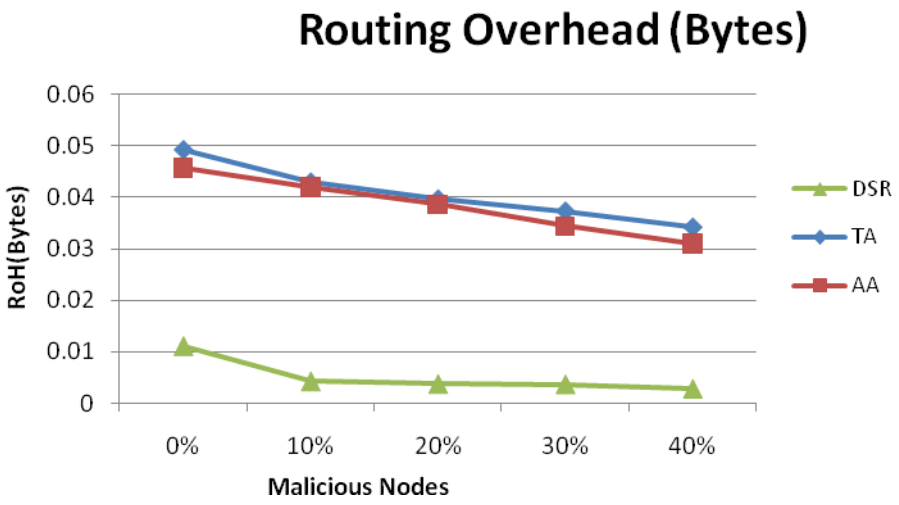


Figure 4-11: Routing overhead of MPEG4 traffic

The last figure, Figure 4.12, has the number of hops in the x axis and the end to end delay in y axis. Normally, it shows that the DSR is the best. Furthermor, it illustrates that all the three mechanisms have almost the same end to end delay for the transmissions of one hop, that is because the AACK and TWOACK schemes do not use any acknowledgments for transmissions of one hop. Also, it shows that both mechnisms have

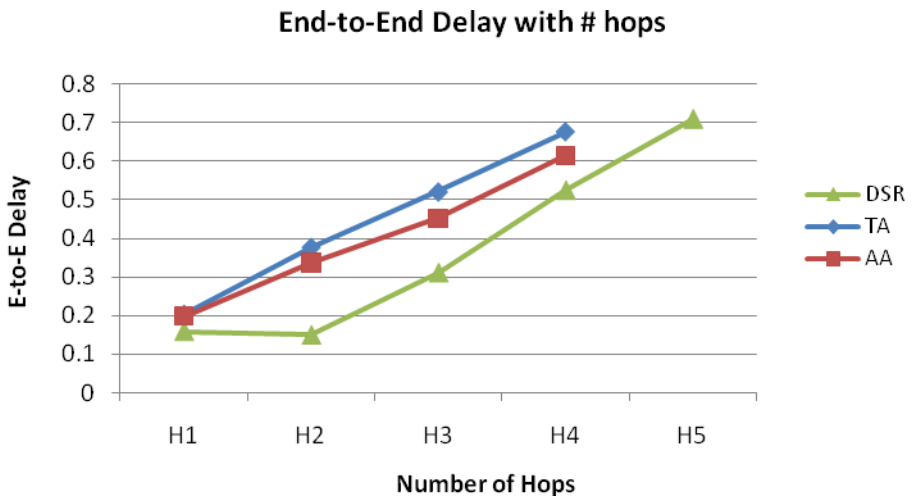


Figure 4-12: End-to-end delay per number of hops

the same delay for 2 hops transmissions, that is because the AACK works as TWOACK for transmissions of 2 hops.

5 CONCLUSIONS AND FUTURE WORK

This chapter summarizes of the whole of this research, reviews the research contributions, and discusses the important future work.

5.1 Conclusions

Intended packet dropping misbehaving could be done by selfish or malicious nodes. This research is devoted to detect and mitigate those misbehaving nodes by avoiding them in later transmissions. In this research we continuo the improvement of the existing IDSs over MANETs, exactly we solve some problems of Watchdog technique, which considered to be the base technique that is used by many of the recently IDSs. Receiver collision and limited power transmission are the two main problems that we focused on to solve in this research. AACK is the proposed IDS, which is an abbreviation for Adaptive ACKnowledgments. AACK is compared to the existing IDS TWOACK to evaluate its performance. The results show that AACK outperforms TWOACK in terms of packet delivery ration and routing overhead in low and high mobility scenarios. Video traffic is used to evaluate AACK IDS, for our best of knowledge this is the first attempt to evaluate IDS with video traffic, the results also shows that AACK is better than TWOACK.

NS-2 is used to conduct our simulations, and we have modified the original DSR protocol to implement the packet dropping attack and the IDSs. This research has several contributions that will be summarized as follows:

- Studied the effect of intended packet dropping misbehavior on mobile ad hoc networks using simulations.

- Proposed new IDS for MANETs, which solve the two problems of watchdog technique, receiver collision and limited power transmission, and improves the performance of existing mechanisms (TWOACK and Watchdog).
- Compared intrusion detection mechanisms in various wireless scenarios
- Examining the MPEG4 and H.246 video traffic over DSR routing protocol.
- Examining video traffic over intrusion detection systems in MANETs.

The implementation of intrusion detection systems over MANETs is not easy especially when it works in high mobility or high traffic load scenarios. It needs to be adaptive to any changes in the network either speed or traffic. I suggest making the timeout and thresholds parameters adaptive to make the IDS more efficient.

The AACK mechanism has some limitations, such as that it could not work will in long paths that will take a significant time for the end to end acknowledgments to be applied. This will give the misbehaving nodes more time for dropping more packets. Also still suffer from the partial dropping attacks (gray hole attacks).

5.2 Future work

- In our future work, AACK will be evaluated in more complicated scenarios such as 100 mobile nodes. Furthermore, we planned to adapt it to work with other MANET routing protocols, not just over DSR. One of the important future works is to study MANET IDS performance under other popular routing protocols (both reactive and proactive).

- This scheme can be used to solve other problems of watchdog like partial dropping or colluding, where two nodes cooperate to do the misbehaving. We are looking for improving AACK to detect the gray hole attacks, which adapt their dropping rates to the IDS' threshold.

- Because most of the previous IDSs mechanisms uses watchdog, we argue that AACK scheme is the most power aware intrusion detection mechanism. That is because it does not use watchdog that uses overhearing, which consume large amount of power. Also, TWOACK does not use watchdog but still consume more power in computations. One of the more important issues of our future work is to evaluate AACK with taken in our consideration the power consumption performance metric.

REFERENCES

[Akyildiz2002] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393-422, March 2002.

[Awerbuch2004] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "Mitigating byzantine attacks in ad hoc wireless networks," Center for Networking and Distributed Systems , Johns Hopkins University," Technical Report, 2004.

[Axelsson2003] S. Axelsson, "Intrusion Detection Systems: A Taxonomy and Survey," Tech. report no. 99-15, Dept. of Comp. Eng., Chalmers Univ. of Technology, Sweden, Mar. 20, 2003.

[Balakrishnan2005] Balakrishnan, K.; Jing Deng; Varshney, V.K., "TWOACK: preventing selfishness in mobile ad hoc networks," *Wireless Communications and Networking Conference, 2005 IEEE* , vol.4, no., pp. 2137-2142 Vol. 4, 13-17 March 2005

[Barbeau2007] M. Barbeau, E. Kranakis, *Principles of Ad-hoc Networking*. Wiley, 2007.

[Broch2002] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad-hoc Networks," *IETF Internet Draft,* Feb. 2002, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt.

[Brutch2003] Brutch, Paul and Calvin Ko, "Challenges in Intrusion Detection for Wireless Ad-Hoc Networks." *Proceedings of Symposium on Applications and the internet Workshop*, 2003.

[Buchegger2002] S. Buchegger and J.-Y. L. Boudec, “Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)," in MOBIHOC'02, 2002.

[Buchegger2005] S. Buchegger and J.-Y. L. Boudec, “Self-policing mobile ad hoc networks by reputation systems," in IEEE Communications Magazine, ser. 7, vol. 43, July 2005, pp. 101-107.

[Buttyan2003] L. Buttyan and J. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks," in Mobile Networks and Applications, 2003, pp. 579-592.

[Clausen2003] T. Clausen, P. Jacquet RFC 3626 - Optimized Link State Routing Protocol (OLSR), http://tools.ietf.org/html/rfc3626, 2003.

[Chee2007] Chee-Onn Chow, Hiroshi Ishii, “Enhancing real-time video streaming over mobile ad hoc networks using multipoint-to-point communication,” Computer Communications, pp. 1754-1764, 2007.

[Chih2008] Chih-Heng Ke, Ce-Kuen Shieh, Wen-Shyang Hwang, Artur Ziviani, “An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission” Journal of Information Science and Engineering 24, 425-440, 2008.

[Chun2003] Y.-C. Hu, A. Perrig, and D. Johnson, “Packet leashes: A defense against wormhole attacks in wireless networks," in the 22nd IEEE Computer and Communications Societies (INFOCOM'03), 2003.

[Chun2-2003] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Rushing attacks and defense in wireless ad hoc network routing protocols," in Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe 2003). ACM, September 2003, pp. 30-40.

[Debar2000] H. Debar, M. Dacier, and A.Wespi, “A Revised Taxonomy for Intrusion Detection Systems,” *Annales des Telecommunications,* vol. 55, pp. 361-378, 2000.

[Frank2004] M. Frank, P. Martini, and M. Plaggemeier, “Cinema: Cooperation enhancement in manets," in Proceedings of the 29th Annual IEEE International Conference on Local Computers Networks LCN'04), 2004.

[Kurkowski2005] S. Kurkowski, T. Camp, and M. Colagrosso. “MANET simulation studies: The current stated and new simulation tools,” Technical report, Department of Math. And Computer Science, Colorado School of Mines, MCS-05-02, February 2005.

[Hasswa2005] Hasswa, A.; Zulkernine, M.; Hassanein, H., "Routeguard: an intrusion detection and response system for mobile ad-hoc networks," *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on* , vol.3, no., pp. 336-343 Vol. 3, 22-24 Aug. 2005

[Hekmat2006] R. Hekmat, *Ad-hoc Networks: Fundamental Properties and Network Topologies*, Springer, 2006.

[Huang2004] Y. an Huang and W. Lee, “Attack analysis and detection for ad hoc routing protocols," in Proceedings of The 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004), September 2004.

[Hu2002] Yih-Chun Hu; Johnson, D.B.; Perrig, A., "SEAD: secure efficient distance vector routing for mobile wireless ad-hoc networks," *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on* , vol., no., pp. 3-13, 2002

[Hu2-2002] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad-hoc Networks," *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MobiCom'02),* Atlanta, GA, pp. 12-23, Sept. 23 - 28, 2002.

[Ilyas2002] M. Ilyas, ed., *The Handbook of Ad-hoc Wireless Networks*. CRC Press, December 2002.

[Islam2005] Islam, M.M.; Pose, R.; Kopp, C., "An Intrusion Detection System for Suburban Ad-hoc Networks," *TENCON 2005 IEEE Region 10* , vol., no., pp.1-6, Nov. 2005.

[ITU1996] ITU, Subjective Video quality assessment methods for multimedia applications, ITU-T Recommendations P.910, 1996. P. Symes, Digital Video Compression, McGraw-Hill, USA, 2001.

[Joa-Ng1999] M. Joa-Ng and I. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad-hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 8, pp. 1415-1425, Aug. 1999.

[Johnson1996] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-hoc Wireless Networks," *Mobile Computing, T. lmielinski and H. Korth, Eds., Kluwer,* 1996, pp. 153-81

[Johnson2004] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, “The dynamic source routing protocol for mobile ad hoc networks (dsr)," Published Online, IETF MANET Working Group, INTERNET-DRAFT, July 2004, expiration: January 2005. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt

[Larsson1998] Tony Larsson, Nicklas Hedman, "Routing Protocols in Wireless Ad-Hoc Networks- A Simulation Study,*" Master's thesis in Computer Science and Engineering, Lulea University of Technology*, 1998.

[Li2004] Li Y., Wei J., "Guidelines on Selecting Intrusion Detection Methods in MANET," *Proc of ISECON* 2004, v 21.

[Lie2007] A. Lie, and J. Klaue,"Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video", Multimedia Systems, 10.1007/s00530-007-0110-0, 13. November 2007.

[Marti2000] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad-hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00),* PP. 255-265, August 2000.

[Mattrawy2002] Matrawy, A.; Lambadaris, L.; Changcheng Huang, "MPEG4 traffic modeling using the transform expand sample methodology," Networked Appliances, 2002. Gaithersburg. Proceedings. 2002, IEEE 4th International Workshop on , vol., no., pp.249-256, 2002.

[Michiardi2002] P. Michiardi and R. Molva, “CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in CMS'2002, Communication and Multimedia Security 2002 Conference, September 26-27, 2002.

[Mishra2004] Mishra, A.; Nadkarni, K.; Patcha, A., "Intrusion detection in wireless ad-hoc networks," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol.11, no.1, pp. 48-60, Feb 2004

[Mukherjee1994] Mukherjee, B.; Heberlein, L.T.; Levitt, K.N., "Network intrusion detection," *Network, IEEE* , vol.8, no.3, pp.26-41, May/Jun 1994.

[Mundinger2005] J. Mundinger and J.-Y. L. Boudec, “Analysis of a reputation system for mobile ad-hoc networks with liars," wiopt, vol. 00, pp. 41-46, 2005.

[Murthy1996] Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” *ACM Mobile Networks and Applications Journal*, vol. 1, no. 2, pp.183-197, 1996.

[Nasser2007] Nasser, N.; Chen, Y., "Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad-hoc Networks," *Communications, 2007. ICC '07. IEEE International Conference on* , vol., no., pp.1154-1159, 24-28 June 2007.

[NS2] “The Network Simulator (ns-2),” http://www.isi.edu/nsnam/ns/, version ns2.33.

[Pandian2006] Pandian, R.; Seethalakashmi, P.; Ramachandran, V., “Enhanced Routing Protocol for Video Transmission over Mobile Adhoc Network,” Jornal of Applied Sciences Research 2(6): 336-340, 2006.

[Patcha2003] Patcha, A.; Mishra, A., "Collaborative security architecture for black hole attack prevention in mobile ad-hoc networks," *Radio and Wireless Conference, 2003. RAWCON '03. Proceedings*, vol., no., pp. 75-78, 10-13 Aug. 2003.

[Patwardhan2005] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure routing and intrusion detection in ad hoc networks," in 3rd International Conference on Pervasive Computing and Communications, March 2005.

[Papadimitratos2002] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad-hoc Networks," *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'02),* San Antonio, TX, pp. 27-31, Jan. 2002.

[Park1997] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of the 9th IEEE International Conference on Computer Communications and Networking (INFOCOM'97), Kobe, Japan, pp. 1405-1413, Apr. 1997.

[Parker2004] Parker, J.; Undercoffer, J.; Pinkston, J.; Joshi, A., "On intrusion detection and response for mobile ad-hoc networks," *Performance, Computing, and Communications, 2004 IEEE International Conference on* , vol., no., pp. 747-752, 2004

[Perkins1994] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM Computer Communication Review,* vol. 24, London, UK, pp. 234-244, Oct., 1994.

[Perkins1999] C. E. Perkins, and E. M. Royer, “Ad-hoc On-Demand Distance Vector (AODV) Routing,” Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’99), New Orleans, LA, pp. 90-100, Feb. 1999.

[Perkins2000] C. E. Perkins, *Ad-hoc Networking*. Addison Wesley Professional, December 2000.

[Raghavan2003] B. Raghavan and A. C. Snoeren, “Priority forwarding in ad hoc networks with self-interested parties," in Workshop on Economics of Peer to Peer, June 2003.

[Sarkar2008] S. K. Sarkar, T. G. Basavaraju, C. Puttamadappa, Ad Hoc Mobile Wireless Networks. Auerbach Publications, 2008.

[Sun2004] Bo Sun, INTRUSION DETECTION IN MOBILE AD-HOC NETWORKS, A Doctor of Philosophy Dissertation, Texas A&M University, May 2004.

[Symes2001] P. Symes, Digital Video Compression, McGraw-Hill, USA, 2001.

[Tanapat 2008] Tanapat Anusas-amornkul, “ON DETECTION MECHANISMS AND THEIR PERFORMANCE FOR PACKET DROPPING ATTACK IN AD HOC NETWORK,” PHD dissertation, university of Pittsburgh, July 24, 2008.

[Xiao2006] Y. Xiao, X. Shen, and D.-Z. Du (Eds.), "A Survey on Intrusion Detection in Mobile Ad-hoc Networks," *Wireless/Mobile Network Security*, pp. 170-196, 2006.

[Yi 2005] P. Yi., Y. Jiang., Y. Zhong., and S. Zhang, "Distributed Intrusion Detection for Mobile Ad Hoc Networks," Proceedings of the 2005 Symposium on Applications and the Internet Workshops (SAINT-W'05), IEEE, 2005.

[Zapata2002] M. G. Zapata and N. Asokan,"Securing ad-hoc routing protocols," in the 2002 ACM Workshop on Wireless Security (WiSe 2002), September 2002, pp. 1-10.

[Zhang2000] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'00)*, Boston, MA, PP. 275-283, Aug. 2000.

[Zhang2003] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, Vol. 9, No. 5, September 2003.

[Zhang2004] Y. Zhang, W. Lou, and Y. Fang, "Sip: a secure incentive protocol against selfishness in mobile ad hoc networks," in IEEE Wireless Communications and Networking Conference (WCNC'04), March 2004.

[Zhou1999] L. Zhou, and Z. Haas, "Securing Ad-hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, November/December, 1999.

APPENDIX A - Pseudo code of the AACK scheme

A. Source Node

```
//initialization at the sender node

Cmis (for all Nodes) =0          //all nodes start as well-behaving node

 Nmode=AA   //initialization mode (default mode AACK)

While (current time < simulation time) do

If (data PKT sent) then    //PKT == packet

        If (Nmode== AA) then PKTtype = AA    // Mark data PKT as TACK PKT

        Else     PKTtype = AA                    //Mark data PKT as AACK PKT

        LIST ←PKT_ID, PKTtype, Ts   // insert PKT ID, PKT type and sending time into
LIST

End

If (AACK OR TACK PKT received) then    //Acknowledgment received

        Search LIST for PKT_ID carried by AACK or TACK

If (found) then

         LIST→PKT_ID   // remove PKT record (PKT_ID, PKTtype, Ts) from LIST
```

End

If (timeout event happens) **then** // AACK or TACK is not received for PKT_ID

If (PKTtype= AA) **then**

Begin

Nmode=TA //switch node mode to Twoack

LIST→PKT_ID // remove PKT record (PKT_ID, PKTtype, Ts) from LIST

End

If (PKTtype =TA) then

LIST→PKT_ID // remove PKT record (PKT_ID,PKTtype, Ts) from LIST

Cmis++ //increase the misbehavior counter of the link

If (Cmis >Thmis) then Report misbehavior link // misbehavior counter exceeds predefined threshold

End

End

If (switch PKT received) **then**

Nmode=AA // switch node mode to AACK

End

B. Forwarder Node

While (current time < simulation time) **do**

If (data PKT received) **then**

If (PKTtype = AA) **then** just forward // current mode is AACK

Else // current mode is TWOACK

Begin

LIST ←PKT_ID, Ts // insert PKT ID and sending time into LIST

Send (TWOACK PKT) // to 2 hops before

End

If (TACK PKT received) **then**

Do same as sender

If (timeout event happens) **then**

Do same as sender when (PKTtype =TA)

End

C. Receiver Node:

While (current time < simulation time) **do**

If (data PKT received) **then**

If (PKTtype = AA) **then**

Send (AACK PKT) //to Sender

Else

Send (Switch PKT) // to Sender

End

Appendix B – Results Tables

A. CBR TRAFFIC

DSR- Low Speed

MN %	0%	10%	20%	30%	40%
PDR	0.99858	0.81194	0.74692	0.70356	0.66637
RoH(PKT)	0.01476	0.01228	0.01139	0.01008	0.00941
RoH(Byt)	0.00296	0.00246	0.00231	0.00205	0.00192
E-E delay	0.01455	0.01197	0.01078	0.01023	0.00976

DSR High Speed

MN %	0%	10%	20%	30%	40%
PDR	0.99581	0.71002	0.62483	0.56137	0.52591
RoH(PKT)	0.11478	0.0762	0.05666	0.04795	0.03986
RoH(Byt)	0.02321	0.01477	0.01083	0.00931	0.00781
E-E delay	0.02884	0.01565	0.01235	0.01053	0.0094

Watchdog Low Speed

MN%	0%	10%	20%	30%	40%
PDR	0.99853	0.87657	0.81362	0.76199	0.6941
RoH(PKT)	0.03639	0.01241	0.01473	0.01892	0.00976
RoH(Byt)	0.00723	0.00246	0.00292	0.00372	0.00196
E-E delay	0.01752	0.01291	0.01221	0.01189	0.01033

Watchdog High Speed

MN%	0%	10%	20%	30%	40%
PDR	0.99581	0.71928	0.64693	0.58548	0.55108
RoH(PKT)	0.11478	0.08468	0.06321	0.05755	0.04279
RoH(Byt)	0.02321	0.01657	0.01216	0.011	0.00816
E-E delay	0.02884	0.01584	0.01307	0.01153	0.0098

TWOACK Low Speed

MN%	0%	10%	20%	30%	40%
PDR	0.99579	0.98328	0.97541	0.90383	0.89024
RoH(PKT)	0.46366	0.47112	0.45274	0.50388	0.49534
RoH(Byt)	0.11829	0.12334	0.11317	0.16532	0.16
E-E delay	0.0196	0.01997	0.0189	0.01874	0.01807

TWOACK High Speed

MN%	0%	10%	20%	30%	40%
PDR	0.71913	0.68576	0.67825	0.5967	0.66051
RoH(PKT)	0.72185	0.71132	0.70397	0.74042	0.6175
RoH(Byt)	0.38686	0.38097	0.35651	0.40974	0.28179
E-E delay	0.05674	0.05206	0.0491	0.04325	0.03449

AACK-Link Detection LS

MN%	0%	10%	20%	30%	40%
PDR	0.99871	0.98639	0.978244	0.96159	0.95211
RoH(PKT)	0.42111	0.43796	0.43659	0.45333	0.47407
RoH(Byt)	0.10159	0.10891	0.10735	0.11776	0.12751
E-E delay	0.018	0.01809	0.01698	0.01824	0.01748

AACK-Link Detection HS

MN%	0%	10%	20%	30%	40%
PDR	0.71856	0.70368	0.67344	0.58466	0.57608
RoH(PKT)	0.73027	0.71823	0.72114	0.75383	0.76127
RoH(Byt)	0.38832	0.37872	0.37744	0.43513	0.41952
E-E delay	0.06424	0.05746	0.05456	0.04238	0.04363

AACK-Node Detection LS

MN%	0%	10%	20%	30%	40%
PDR	0.99843	0.99031	0.9774	0.96159	0.94931
RoH(PKT)	0.42122	0.43381	0.44069	0.45333	0.45226
RoH(Byt)	0.10197	0.10676	0.10895	0.11776	0.11846
E-E delay	0.01792	0.01828	0.01774	0.01824	0.01659

AACK-Node Detection HS

MN%	0%	10%	20%	30%	40%
PDR	0.94162	0.79437	0.73118	0.72413	0.6223
RoH(PKT)	0.59009	0.63562	0.63159	0.59814	0.62961
RoH(Byt)	0.20876	0.27281	0.28959	0.23973	0.28983
E-E delay	0.06543	0.05691	0.04897	0.03301	0.03076

B. VIDEO TRAFFIC

DSR

MN %	0%	10%	20%	30%	40%
PDR	0.34001	0.31061	0.30439	0.29683	0.28492
ROH	0.0681	0.02774	0.0247	0.02391	0.01818
ROH	0.01108	0.00421	0.00371	0.00366	0.00275
E-E delay	0.18787	0.20515	0.2402	0.22247	0.2051

TWOACK

MN%	0%	10%	20%	30%	40%
PDR	0.33765	0.31212	0.31772	0.31137	0.28995
ROH	0.27552	0.2549	0.24187	0.22758	0.21779
ROH	0.04921	0.04292	0.03963	0.03728	0.03415
E-E Delay	0.21959	0.22306	0.24031	0.25743	0.28402

AACK

MN%	0%	10%	20%	30%	40%
PDR	0.33063	0.32343	0.31542	0.32052	0.31404
ROH	0.26102	0.24884	0.23544	0.21832	0.19442
ROH	0.04573	0.04198	0.03864	0.03439	0.03107
E-E Delay	0.21782	0.21185	0.20885	0.221	0.22427

VITA

Anas Abdulwahed Al-Roubaiey was born on 22 May 1976, at Ibb, Yemen. He obtained his Bachelor of Science (BS) degree with honors in computer engineering from AAST University, Egypt in February 2001. Prior to attending King Fahd University of Petroleum & Minerals (KFUPM), he worked as a lecturer from September 2001 to Jun 2004 in Taiz University (Computer Science Department). He joined KFUPM as a full time student to pursue the master's degree. He received Master of Science (MS) degree in Computer Networks from KFUPM in Jun 2009.